# Course Curriculum
## (Course Structure and Syllabi)
## for
## Bachelor of Technology
### in
## *Computer Science and Engineering*

### *(Second Year Onwards)*



**Department of Computer Science and Engineering**

# National Institute of Technology Hamirpur

## Hamirpur – 177 005 (India)

| | | | Second Year | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 3rdSemester | | | | | | | 4thSemester | | | | | |
| SN | Code | Subject | L | T | P | Credits | SN | Code | Subject | L | T | P | Credits | |
| 1 | MA-203 | Engineering Mathematics-III | 3 | 1 | 0 | 4 | 1 | HS-203 | Organizational Behaviour | 3 | 0 | 0 | 3 | |
| 2 | CS-211 | Object Oriented Programming | 3 | 1 | 0 | 4 | 2 | CS-201 | Data Structures | 3 | 1 | 0 | 4 | |
| 3 | CS-212 | Microprocessor and Interfacing | 3 | 0 | 0 | 3 | 3 | CS-221 | Computer Organization and Architecture | 3 | 1 | 0 | 4 | |
| 4 | CS-213 | Discrete Structures | 3 | 1 | 0 | 4 | 4 | CS-222 | Operating System | 3 | 1 | 0 | 4 | |
| 5 | EC-211 | Digital Electronics and Logic Design | 3 | 1 | 0 | 4 | 5 | CS-223 | Theory of Computation | 3 | 1 | 0 | 4 | |
| 6 | CS-214 | Object Oriented Programming Lab | 0 | 0 | 2 | 1 | 6 | CS-202 | Data Structures Lab | 0 | 0 | 2 | 1 | |
| 7 | CS-215 | Microprocessor and Interfacing Lab | 0 | 0 | 2 | 1 | 7 | CS-224 | Computer Organization and Architecture Lab | 0 | 0 | 2 | 1 | |
| 8 | EC-214 | Digital Electronics and Logic Design Lab | 0 | 0 | 2 | 1 | 8 | CS-225 | Operating System Lab | 0 | 0 | 2 | 1 | |
| | | Total Hours = 25 | | | | 22 | | | Total Hours = 25 | | | | 22 | |

| | | | Third Year | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 5thSemester | | | | | | | 6thSemester | | | | | |
| SN | Code | Subject | L | T | P | Credits | SN | Code | Subject | L | T | P | Credits | |
| 1 | CS-311 | Analysis and Design of Algorithms | 3 | 1 | 0 | 4 | 1 | CS-321 | Distributed Systems | 3 | 1 | 0 | 4 | |
| 2 | CS-312 | Data Base Management Systems | 3 | 1 | 0 | 4 | 2 | CS-322 | Software Engineering | 3 | 1 | 0 | 4 | |
| 3 | CS-313 | Compiler Design | 3 | 1 | 0 | 4 | 3 | CS-323 | Digital Image Processing | 3 | 1 | 0 | 4 | |
| 4 | CS-314 | Computer Graphics | 3 | 0 | 0 | 3 | 4 | CS-324 | Computer Networks | 3 | 0 | 0 | 3 | |
| 5 | OET | Open Elective-I | 3 | 0 | 0 | 3 | 5 | OET | Open Elective-II | 3 | 0 | 0 | 3 | |
| 6 | CS-315 | Data Base Management Systems Lab | 0 | 0 | 2 | 1 | 6 | CS-325 | Digital Image Processing Lab | 0 | 0 | 2 | 1 | |
| 7 | CS-316 | Compiler Design Lab | 0 | 0 | 2 | 1 | 7 | CS-326 | Computer Networks Lab | 0 | 0 | 2 | 1 | |
| 8 | CS-317 | Computer Graphics Lab | 0 | 0 | 2 | 1 | 8 | CS-329 | Seminar | 0 | 0 | 2 | 1 | |
| | | Total Hours = 24 | | | | 21 | | | Total Hours = 24 | | | | 21 | |

| Fourth Year | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7thSemester | | | | | | | 8thSemester | | | | | | |
| SN | Code | Subject | L | T | P | Credits | SN | Code | Subject | L | T | P | Credits |
| 1 | HS-404 | Engineering Economics and Accountancy | 3 | 0 | 0 | 3 | 1 | CS-421 | Dataware Housing and Data Mining | 3 | 0 | 0 | 3 |
| 2 | CS-411 | Artificial Intelligence | 3 | 0 | 0 | 3 | 2 | CS-422 | Information Security | 3 | 0 | 0 | 3 |
| 3 | DET | Professional Elective-I | 3 | 0 | 0 | 3 | 3 | DET | Professional Elective-III | 3 | 0 | 0 | 3 |
| 4 | DET | Professional Elective-II | 3 | 0 | 0 | 3 | 4 | DET | Professional Elective-IV | 3 | 0 | 0 | 3 |
| 5 | CS-418 | Industrial Training Presentation | 0 | 0 | 2 | 1 | 5 | CS-428 | General Proficiency | 0 | 0 | 0 | 1 |
| 6 | CS-419 | Major Project (Stage-I) | 0 | 0 | 12 | 6 | 6 | CS-429 | Major Project (Stage-II) | 0 | 0 | 12 | 6 |
| | | Total Hours = 26 | | | | 19 | | | Total Hours = 24 | | | | 19 |

| Semester Wise Credits | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Semester | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th | | Total |
| Credits | 24 | 24 | 22 | 22 | 21 | 21 | 19 | 19 | | 172 |
| Hours/week | 28 | 28 | 25 | 25 | 24 | 24 | 26 | 24 | | 204 |

# Professional Elective Courses

**Professional Elective-I**  (courses related to tools/techniques)

| | |
|---|---|
| CS-430 | Modeling and Simulation Techniques |
| CS-431 | Soft Computing Methods in Engineering |
| CS-432 | Data Analytics |

**Professional Elective-II**  (courses related to Computer Networking)

| | |
|---|---|
| CS-450 | Cyber Forensics |
| CS-451 | Wireless Sensor Networks |
| CS-452 | Next Generation Networks |

**Professional Elective-III**  (courses related to Artificial Intelligence)

| | |
|---|---|
| CS-440 | Information Retrieval |
| CS-441 | Humanoid Robotics |
| CS-442 | Natural Language Processing |

**Professional Elective-IV**  (courses related to Computer Architecture)

| | |
|---|---|
| CS-460 | Reconfigurable Computing |
| CS-461 | Cloud Computing |
| CS-462 | Advanced Computer Architecture |

# Open Elective Courses

**Open Elective-I**

| | |
|---|---|
| CS-306 | Data Structures |
| CS-370 | Operating System |

**Open Elective-II**

| | |
|---|---|
| CS-306 | Data Structures |
| CS-380 | Computer Networks |

| Course Name: | **Engineering Mathematics-III** | |
|---|---|---|
| Course Code: | **MA-203** | |
| Course Type: | **Core** | |
| Contact Hours/Week:  **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To introduce the fundamental concepts relevant to function of complex variable, numerical differentiation and integration and numerical solution of linear, non-linear and system of equations.
- To have the idea of evaluation of real integrals using complex variable.
- To understand the concept of approximating & interpolating polynomials and finding values of function at arbitrary point.
- To impart knowledge of various numerical technique to solve ODE.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Functions of Complex Variable**<br>Applications of De Moivre's theorem, Exponential, Circular, Hyperbolic and Logarithmic functions of a complex variable, Inverse Hyperbolic functions, Real and imaginary parts of Circular and Hyperbolic functions, Summation of the series- 'C+iS' method.<br>Limit and derivative of complex functions, Cauchy-Riemann equations, Analytic functions and its applications, Complex integration, Cauchy's theorem, Cauchy's integral formula, Series of complex function, Taylor series, singularities and Laurent's series, Cauchy's residue theorem and its application for the evaluation of real definite integrals. | **12 L** |
| UNIT-02 | **Interpolation**<br>Least square curve fit and trigonometric approximations, Finite differences and difference operators, Newton's interpolation formulae, Gauss forward and backward formulae, Sterling and Bessel's formulae, Lagrange's interpolation. | **06L** |
| UNIT-03 | **Numerical Integration**<br>Integration by trapezoidal and Simpson's rules 1/3 and 3/8 rule, Romberg integration, and Gaussian quadrature rule, Numerical integration of function of two variables. | **05L** |
| UNIT-04 | **Numerical Solution of Ordinary Differential Equations**<br>Taylor series method, Picard's method, Euler's method, Modified Euler's method, Runge- Kutta method. Predictor corrector methods, Adam Bashforth and Milnes method, convergence criteria, Finite difference method. | **07L** |
| UNIT-05 | **Numerical Solution of Linear and Non-Linear Equations**<br>Non-Linear Equations: Bisection Method, Regula Falsi Method, Newton-Raphson Method, Iteration method. Linear Equations: Jacobi and Gauss Seidal Iteration methods, Relaxation method. | **06 L** |

**Course Outcomes**

Upon successful completion of the course, the student will be able to

CO1: Understand and analyze the concept of Numerical Solution of Linear and Non-Linear Equations, Ordinary Differential Equations and Function of complex variable.

CO2: Identify an appropriate technique to solve the linear, non-linear equations, ordinary differential equations.

CO3: Formulate the problems on related topics and solve analytically.

CO4: Apply the concepts of linear, non-linear equations, differential equations and complex analysis in various        engineering problems.

CO5: Demonstrate the concepts through examples and applications.

**Books and References**

1. Complex variables and Applications by R. V. Churchill, T. J. Brown & R. F. Verhey, McGraw Hill.
2. A first course in complex analysis with applications by Dennis D. Zill and  P. D. Shanahan, Jones and Bartlett.
3. Numerical Methods for Scientific and Engineering Computations by M. K. Jain, S. R. K. Iyenger and R. K. Jain, New Age International Publishers, New Delhi.
4. Numerical Methods for Engineers and Scientists by J D Hoffman, CRC Press.
5. Numerical Analysis Mathematics and Scientific computing by D. Kincaid and W. Cheney, American Mathematical Society.

| Course Name: | **Object Oriented Programming** | |
|---|---|---|
| Course Code: | **CS-211** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the concept of Object Oriented programming.
- To introduce the fundamental concepts relevant to Arrays, Pointers and Functions, Classes, Objects, etc.
- To enable the students to understand the standard library, exception handling, streams and files.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Concepts of Object-Oriented Programming:** Object Oriented Programming Paradigm, Basic concepts of OOPs, Benefits of OOPs, and Introduction to object oriented design and development, Design steps, Design example, Object oriented languages, Comparison of structured and object-oriented programming languages. | **06L** |
| UNIT-02 | **Arrays, Pointers and Functions:** Arrays, Storage of arrays in memory, Initializing Arrays, Multi-Dimensional Arrays, Pointers, accessing array elements through pointers, passing pointers as function arguments, Arrays of pointers, Pointers to pointers, Functions, Arguments, Inline functions, Function Overloading Polymorphism**.** | **06L** |
| UNIT-03 | **Classes and Objects:** Data types, operators, expressions, control structures, arrays, strings, Classes and objects, access specifiers, constructors, destructors, operator overloading, type conversion. <br><br> Storage classes: Fixed vs. Automatic declaration, Scope, Global variables, register specifier, Dynamic memory allocation. | **07L** |
| UNIT-04 | **Inheritance:** Inheritance, single Inheritance, Multiple Inheritance, Multi-level inheritance, hierarchical inheritance, hybrid inheritance, Virtual functions. <br><br> Streams and Files: Opening and closing a file, File pointers and their manipulations, Sequential Input and output operations, multi-file programs, Random Access, command line argument, string class, Date class, Array class, List class, Queue class, User defined class, Generic Class. | **07L** |
| UNIT-05 | **Exception Handling and Graphics:** List of exceptions, catching exception, handling exception. Text Mode, Graphics mode functions, Rectangles, and Lines, Polygons and Inheritance, Sound and Motion, Text in Graphics Mode. <br><br> Standard Template Library**:** Standard Template Library, Overview of Standard Template Library, Containers, Algorithms, Iterators, Other STL Elements, Container Classes, General Theory of Operation, Vectors**.** | **10L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Understand the concept of object oriented paradigm and programming.

CO2: Apply the concept of polymorphism and inheritance.

CO3: Implement exception handling and templates.

CO4: Handling of files and streams during programming.

**Books and References**

1. Object Oriented programming with C++ by E. Balagurusamy, Tata McGraw Hill.
2. The C++ programming Language by Bjarne Strustrup, Addison Wesley.
3. Object Oriented Analysis and Design with Applications by Grady Booch, Addison Wesley.
4. The Complete Reference Visual C++ by Chris H. Pappas and William H. Murray, Tata McGraw Hill.
5. C++ Primer by S. B. Lippman, Josee Lajoie and Barbara E. Moo, Pearson Education.

| Course Name: | **Microprocessor and Interfacing** | |
|---|---|---|
| Course Code: | **CS-212** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about microprocessors and internal architecture of 8085/8086 microprocessors.
- To introduce the fundamental concepts relevant to assembly language programs to solve a given problem.
- To enable the students to understand and implement programs to interface the 8085 microprocessor with peripheral devices.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction to Microprocessor:** History and Evolution, types of microprocessors, 8085 Microprocessor, Architecture, Bus Organization, Registers, ALU, Control section, Instruction set of 8085, Instruction format, Addressing modes, Types of Instructions. | **07L** |
| UNIT-02 | **Assembly Language Programming and Timing Diagram:** Assembly language programming in 8085, Macros, Labels and Directives, Microprocessor timings, Microinstructions, Instruction cycle, Machine cycles, T states, State transition diagrams, Timing diagram for different machine cycles. | **07L** |
| UNIT-03 | **Serial I/O, Interrupts and Comparison of Contemporary Microprocessors:** Serial I/O using SID, SOD. Interrupts in 8085, RST instructions, Issues in implementing interrupts, Multiple interrupts and priorities, Daisy chaining, Interrupt handling in 8085, Enabling, disabling and masking of interrupts. Brief comparison of contemporary 8-bit microprocessors like Z-80, M68000 with 8085. | **06L** |
| UNIT-04 | **Data Transfer techniques:** Data transfer techniques, Programmed data transfer, Parallel data transfer using 8155. Programmable parallel ports and handshake input/output, Asynchronous and Synchronous data transfer using 8251A. Programmable interrupt controller 8259A. DMA transfer, cycle stealing and burst mode of DMA, 8257 DMA controller. | **06L** |
| UNIT-05 | **Microprocessor Interfacing Techniques:** Interfacing memory and I/O devices, Addressing memory, interfacing static RAMs, Interfacing and refreshing dynamic RAMs, Interfacing a keyboard, Interfacing LED and seven segment displays, interfacing a printer, Interfacing A/D converters, D/A converters.<br><br>Architecture of 8086: Memory Address space and data organization,- segment registers and memory segmentation, Generating memory addresses, IO address space, addressing modes, Comparison of 8086 and 8088, minimum mode maximum mode, system timing, introduction to Pentium and further series of microprocessors. | **10L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Understand the concepts of microcomputer and microprocessors and internal architecture of 8085/8086 microprocessors.

CO2: Write and implement assembly language programs to solve a given problem.

CO3: Write and implement programs to interface the 8085 microprocessor with peripheral devices.

**Books and References**

1. Fundamentals of Microprocessors and Microcomputers by B. Ram, Dhanpat Rai and Sons.
2. Microprocessor Architecture, Programming and applications with the 8085/8080A by R.S. Gaonkar, Wiley.
3. Microprocessors& Interfacing by Douglas V Hall, McGraw Hill.
4. Microprocessors and Digital Systems by Douglas V Hall, McGraw Hill.
5. Introduction to Microprocessor by A.P. Mathur, Tata McGraw Hill.

| Course Name: | **Discrete Structures** | |
|---|---|---|
| Course Code: | **CS-213** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about disjunctive normal form and determine their validity.
- To introduce the fundamental concepts of common language to formal logic using the rules of propositional and predicate calculus.
- To enable the students to understand the logic recursion, basic structure and graph theory.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction to Preliminaries and Predicate Calculus:** Basic concepts of discrete mathematics and related problems**,** propositions and predicates, disjunction and conjunction, tautologies and contradiction, laws of equivalence, rules of substitution and transitivity, normal forms. | **07L** |
| UNIT-02 | **Set Theory and Functions:** Basic concepts, ,Venn Diagrams, set operations, power set, methods of proof for sets, Relations and ordering, Types of relations, Graph and matrix of a relation, properties of a relation, Functions: definitions and notation, one to one, onto, one to one and onto, composition, identity and inverse, related results**.** | **07L** |
| UNIT-03 | **Graph Theory:** Basic concepts of graph theory, multigraphs and weighted graphs, walk path and circuits, Warshall's algorithm: shortest path, Eulerian paths and circuits, Hamiltonian paths and circuits, factors of a graph and planar graphs, Graph colorings. | **06L** |
| UNIT-04 | **Binary Trees:** Introduction, complete and extended binary tree, traversing binary tree, binary search tree, Heaps, Huffman's algorithm. | **06L** |
| UNIT-05 | **Basics of Structures:** Mathematical induction, Algebraic structures properties, Semi group, Monoid, Group and Sub group with examples and standard results, generators and evaluation of powers, cosets and Langranges's theorem, rings, integral domains, fields.<br><br>**Logic and Recursion:** Propositional calculus-propositions, logical operators, truth tables, Lattice, propositions generated by a set of recurrence relations – partial and total recursion - problems. | **10L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Interpret statements presented in disjunctive normal form and determine their validity.
CO2: Reformulate statements from common language to formal logic.
CO3: Understand and apply the properties of relations in computer science and engineering problems.
CO4: Understand graph problems and implement effective solutions.

**Books and References**

1. Discrete Mathematical structures with applications to Computer Science by J. P. Tremblay and R Manohar, McGraw Hill.
2. Elements of Discrete Mathematics by C.L. Liu, McGraw Hill.
3. An Introduction to Discrete Mathematics by Steven Roman, Saunders Publisher.
4. Discrete Mathematics by Steven Barnett, Addison Wesley.
5. Discrete Mathematics by John A. Dossey, Albert D. Otto, Lawrence E. Spence and Vanden Eynden, Charles Addison-Wesley.
6. Discrete Mathematics and Its Applications by Kenneth H. Rosen, McGraw Hill.

| Course Name: | **Digital Electronics and Logic Design** | |
|---|---|---|
| Course Code: | **EC-211** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the concept of digital design, number system and codes
- To introduce the fundamental concepts related to design of combinational logic circuits
- To enable the students to understand the design of Sequential Circuits

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** Analog versus Digital, Analog to Digital and Digital to Analog converter circuits; Number systems and their inter-conversion, Binary Arithmetic (Addition, Subtraction, Multiplication and Division), Diminished radix and radix compliments; BCD codes, Excess-3code, Gray code, Hamming code, Error Detection and Correction. | **07L** |
| UNIT-02 | **Logic Gates and Logic Families:** Digital Logic Gates, Various Logic Families: RTL, DTL, TTL and ECL; Working and their characteristics; MOS and CMOS devices. | **06L** |
| UNIT-03 | **Combinational Logic Design:**Boolean Algebra, Basic Theorems and Properties of Boolean Algebra, Minimization of Logical functions, Karnaugh- Map method, Sum of Products and Product of Sums Simplification, NAND and NOR implementation, Incompletely Specified functions, VEM method, Tabulation method,  Determination of Prime implicants, Selection of Essential Prime implicants, Iterative Consensus & Generalized Consensus method for minimization of Multiple Output Switching functions,  Determination of Prime implicants, Selection of Essential Prime implicants and finding a minimal cover, Design of Combinational circuits with examples. | **10L** |
| UNIT-04 | **MSI and PLD Components:** Binary Adder and Subtractor; Decoders and Encoders; Multiplexers and DE-Multiplexers circuits; Read Only Memory, Programmable Logic Arrays, Programmable Array Logic; Implementation of Combinatorial Logic using these devices. | **06L** |
| UNIT-05 | **Sequential Logic Design:** Introduction and Classification of Sequential circuits, Flip-flops: Truth Table & Excitation Table of flip-flops, Inter conversion of flip-flops, Design of Synchronous & Asynchronous Sequential circuits, Registers and Counters, | **07L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Understand about the concept of digital system

CO2:    Apply principles of minimization techniques to simplify digital functions

CO3:    Design and analyse the combinational electronic circuit based on digital logic

CO4:    Design and analyse the sequential electronic circuit based on digital logic

**Books and References**

1. Digital Design: M. Morris Mano, Prentice Hall of India.
2. Digital Principle and Applications: Malvino and Leach, Tata Mc-Graw Hill.
3. Fundamentals of Digital Electronics: Anand Kumar, Prentice Hall of India.
4. Modern Digital Electronic: R.P.Jain Tata Mc-Graw Hill.

| Course Name: | **Object Oriented Programming Lab** | |
|---|---|---|
| Course Code: | **CS-214** | |
| Contact Hours/Week: **2P** | | Course Credits: **01** |

**Course Objectives**

- To provide skills for designing object oriented programs.
- To provide skills for writing C++ programs.
- To enable the students to debug and implement programs involving different concepts of object oriented paradigm.

**List of Experiments**

1. Demonstrate simple operations and control structures.
2. Demonstrate use of array : i) Add two matrix using multi-dimensional arrays, ii) Searching in arrays, Array of Class, Operations on strings as arrays and to access array elements using pointers.
3. Demonstrate use of structure and union.
4. Demonstrate use of function by i) To swap numbers in cyclic order using call by reference and to check prime number by creating a function, ii) To find the multiplication values and the cubic values using inline function, iii) To calculate the area of circle, rectangle and triangle using function overloading.
5. Read and print students detail using class and object.
6. Demonstrate example of friend function with class.
7. Demonstrate increment ++ and decrement - - variables by using operator overloading.
8. Write a program to demonstrate the single inheritance, multilevel inheritance, multiple inheritances, hybrid inheritance and hierarchical inheritance.
9. Write a program that shows that two files named 'Source 1' and 'Source 2' contains sorted list of integers. Read the content of both the files and stores the merged list in sorted form in a new file named 'Target'.
10. Write a function template to perform linear search in an array.
11. Write a program containing a possible exception. Use a try block to throw it and a catch block to handle it properly.
12. Write a program to demonstrate container i) using the find () algorithm to locate the position of a specified value in a sequence container, ii) using the algorithm count () to count how many elements in a container have a specified value.
13. Write a program to implement Vector in STL.

*Note: The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Identify and abstract the programming task involved for a given problem.

CO2:     Design and develop object oriented programming skills.

CO3:     Trace and debug a program.

| Course Name: | **Microprocessor and Interfacing Lab** | |
|---|---|---|
| Course Code: | **CS-215** | |
| Contact Hours/Week: **2P** | | Course Credits: **01** |

**Course Objectives**

- To impart knowledge about microprocessors and internal architecture of 8085/8086 microprocessors.
- To introduce the fundamental concepts relevant to assembly language programs to solve a given problem.
- To enable the students to understand and implement programs to interface the 8085 microprocessor with peripheral devices.

**List of Experiments**

1. Write a program to transfer an 8 bit data from i) register B to accumulator, ii) memory location to accumulator, iii) to accumulator by using 1) MOV, 2) LDA, 3) LDAX.
2. Write a program to find out the 2's complement of a given i) 8 bit number, ii) 16 bit number.
3. Write a program to find out the numbers of odd and even numbers from an array of numbers.
4. Write a program to i) add two 8 bit numbers considering the carry, ii) subtract two 16 bit numbers considering the borrow, iii) multiply two 8 bit numbers, iv) sum of a series of 8 bit numbers.
5. Write a program to mask the lower nibble of an 8 bit numbers.
6. Write a program to shift an i) 8 bit number left by 2 bit, ii) 16 bit number by 2 bit without using rotate instruction.
7. Write a program to find out the square of an 8 bit using look up table.
8. Find out the smallest and largest from an array of numbers.
9. Sort array of numbers in ascending and descending order.
10. Transfer block of N-bytes from source to destination in forward and reverse order.
11. Write a program to i) Unpack a BCD number, ii) Pack two BCD numbers, iii) convert two BCD numbers in memory location to the equivalent HEX number.
12. Write a program to find out the factorial of a given number.
13. Write a program to interface 8255A with 8085 or 8086 microprocessor kit.
14. Study of 8251 microcontroller kit and Interfacing of keyboard with 8251 microcontroller.

*Note: The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:   Understand the concepts of microcomputer and microprocessors and internal architecture of 8085/8086 microprocessors.
CO2:   Write and implement assembly language programs to solve a given problem.
CO3:   Write and implement programs to interface the 8085 microprocessor with peripheral devices.

| Course Name: | **Digital Electronics and Logic Design Lab** |
| --- | --- |
| Course Code: | **EC-214** |

| Contact Hours/Week: **2P** | Course Credits: **01** |
| --- | --- |

**Course Objectives**

- Familiarization with digital integrated circuits and equipments
- Implementation and design of combinational logic circuits using different gates
- To understand concepts of sequential circuits and to analyze and design sequential circuits

**List of Experiments**

1. To study about the logic gates and verify their truth table.
2. Realization of AND and OR gates using
   (i) Diodes and resistors.
   (ii) Universal gates
3. Design and implement half adder and full adder circuits and verifies the truth table using logic gates.
4. Design and implement half subtractor and full subtractor circuits and verifies the truth table using logic gates.
5. Design and implement 4-bit binary to gray code converter and gray to binary code converter circuits.
6. Design and implement BCD to excess-3 code converter and excess-3 to BCD code converter.
7. Design and implement
   (i) 2-Bit magnitude comparator using basic gates
   (ii) 8-Bit magnitude comparator using IC 7485
8. Design and implement multiplexer and demultiplexer using logic gates and study of IC 74150 and IC 74154.
9. Design and implementation of the function using multiplexer
   (i) $F(A,B,C)=\Sigma m(1,2,5,6)$
   (ii) $F(A,B,C)=\Sigma m(0,2,5,6,7)$
10. Design and implement encoder and decoder using logic gates and study of IC 7445 and IC 74147.
11. Realization of SR, JK, D and T flip flop using gates.
12. Design and implement 3-bit asynchronous up/down counter.

*Note: The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Understand the digital signals, applications of ICs and logic circuits
CO2: Develop skills for designing combinational logic circuits and their practical implementation on breadboard
CO3: Analyze, design and implement sequential logic circuits

| Course Name: | **Organizational Behaviour** | |
|---|---|---|
| Course Code: | **HS-203** | |
| Course Type: | **Core** | |
| Contact Hours/Week:  **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about the behavioral aspects related to professional organizations
- To introduce the fundamental concepts relevant to understanding of individual & group behavior in the organization
- To enable the students to understand the applied organizational themes like perception, motivation, interpersonal relationships, group dynamics, leadership theories, role of power & politics in organizational context, conflict and negotiation, organizational diversity, dynamics of personality, attitude and job satisfaction, etc.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Organizational Behavior (OB):** Concept, nature, characteristics, conceptual foundations, determinants and importance, management functions, role & skills, disciplines that contribute to the field of OB, Challenges & Opportunities for OB, diversity in Organizations, attitudes & Job satisfaction. | **04L** |
| UNIT-02 | **Perception**: Concept, nature, process, importance, management and behavioral applications of perception. Personality: concept, nature, types and theories of personality shaping. Learning; concept and theories of learning. | **08L** |
| UNIT-03 | **Motivation:** concept, principles, theories-content, process & contemporary, Monetary and non-monetary motivation, applications of motivation. Leadership: Concept, functions, styles, and theories of leadership- trait, behavioral, and situational. | **06L** |
| UNIT-04 | **Group and Interpersonal Relationship:** Analysis of Interpersonal Relationship, developing interpersonal relationship, Group Dynamic: Definition of Group, stages of Group Development, Punctuated Equilibrium Model, Group Structure, Group Decision Making, understanding work teams. | **05L** |
| UNIT-05 | **Organizational Power and Politics:** concept of power, structure of power, classification of power, contrasting leadership & power, dependence a key to power, causes & consequences of political behavior. Organizational conflict: view of conflict, conflict process, negotiation & bargaining strategies. | **06L** |
| UNIT-06 | **Conflict and Negotiation:** conflict definition in conflict thought: Traditional view, the Human relation view, interactionist view. Functional versus dysfunctional conflict, conflict process. Negotiation Bargaining strategies, the negotiation process and issues in negotiation. | **07L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Identify the challenges of the present organization

CO2:    Describe the organizational system

CO3:    Apply the principles of organizational behavior in order to inculcate the habit of team work and which is essential for the organization

CO4:    Assess the role of psychological and social principles in improvement of efficiency as well as quality of the employee life

**Books and References**

1.    Robbins, S.P., Organizational Behavior, Prentice Hall of India.
2.    Luthans F., Organizational Behavior, McGraw-Hill.
3.    Davis K., Human Behavior at Work: Organizational Behavior, Tata McGraw-Hill.

| Course Name: | **Data Structures** | |
|---|---|---|
| Course Code: | **CS-201** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about linear and non-linear data structures as the foundational base for computer solutions to problems.
- To introduce the fundamental concepts relevant to binary trees, binary tree traversals, binary search trees and perform related analysis to solve problems.
- To enable the students to understand various types of sorting algorithms.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** Data types, data structures, abstract data types, the running time of a program, the running time and storage cost of algorithms, complexity, asymptotic complexity, big O notation, obtaining the complexity of an algorithm. | **07L** |
| UNIT-02 | **Development of Algorithms:** Notations and Analysis, Storage structures for arrays - sparse matrices - structures and arrays of structures, Stacks and Queues: Representations, implementations and applications.<br><br>Linked Lists: Singly linked lists, Linked stacks and queues, operations on Polynomials, Doubly Linked Lists, Circularly Linked Lists, Operations on linked lists- Insertion, deletion and traversal, dynamic storage management – Garbage collection and compaction. | **10L** |
| UNIT-03 | **Trees:** Basic terminology, General Trees, Binary Trees, Tree Traversing: in-order, pre-order and post-order traversal, building a binary search tree, Operations on Binary Trees - Expression Manipulations - Symbol Table construction, Height Balanced Trees(AVL), B-trees, B+-trees. | **07L** |
| UNIT-04 | **Graphs:** Basic definitions, representations of directed and undirected graphs, the single-source shortest path problem, the all-pair shortest path problem, traversals of directed and undirected graphs, directed acyclic graphs, strong components, minimum cost spanning tress, articulation points and biconnected components, graph matching. | **06L** |
| UNIT-05 | **Sorting and Searching Techniques:** Bubble sorting, Insertion sort, Selection sort, Shell sort, Merge sort, Heap and Heap sort, Quick sort, Radix sort and Bucket sort, Address calculation, Sequential searching, Binary Searching, Index searching, Hash table methods. | **06L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Interpret and compute asymptotic notations of an algorithm to analyze the time complexity.

CO2:    Use of linear and non-linear data structures as the foundational base for computer solutions to problems.

CO3:    Demonstrate the ability to implement various types of static and dynamic lists.

CO4:    Implement binary trees, binary tree traversals, and binary search trees.

CO5:    Implement various types of sorting algorithms.

**Books and References**

1.    An Introduction to Data Structures with applications by J.P. Tremblay and P.G. Sorenson, Tata McGraw Hill.

2.    Data structures, Algorithms ad Applications in C++ by Sartaj  Sahni, WCB/McGraw Hill.

3.    Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman and  John E. Hopcroft, Addison Wesley.

4.    Data Structures using C by Y. Langsam, M. J. Augenstein and A. M. Tenenbaum, Pearson Education.

5.    Data Structures – A Pseudocode Approach with C by Richard F. Gilberg and Behrouz A. Forouzan, Thomson Brooks /Cole.

| Course Name: | **Computer Organization and Architecture** | |
|---|---|---|
| Course Code: | **CS-221** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the organization of any computing system.
- To introduce the fundamental concepts relevant to design instruction set architectures and develop their micro architectures.
- To enable the students to understand the factors various caching and architecture memory system architectures and instruction level parallelism.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **General System Architecture:** Stored Program control concept (Von-Newman architecture principle), Flynn's Classification of computers (SIMD, MISD, MIMD), Structure organization (CPU, Caches, Main memory, Secondary memory unit & I/O), Register Transfer Operation, Micro-operation, Instruction Set Architecture (Instruction set based classification of processor i.e. RISC, CISC), Addressing Modes, Operation instruction set (Arithmetic & logical, Data transfer, Control flow), Instruction set format. | **07L** |
| UNIT-02 | **Processor Design:** Arithmetic and logic unit, Stack organization, CPU Architecture types, Accumulator Based- Register, Stack-Memory, Register, Detailed data path of a typical register based CPU, Fetch, Decode, and Execute Cycle. Addition and Subtraction, Multiplication Algorithms (Booth's Multiplication Algorithm), Division Algorithm, Floating point arithmetic operations. | **08L** |
| UNIT-03 | **Control Design:** Microprogrammed and Hard-wired control options, Hard-wired design methods, State table method, Multiplier control, CPU control unit. Microprogrammed, Basic concepts, control Memory, Address Sequencing | **07L** |
| UNIT-04 | **I/O Organization & Memory Hierarchy:** Programmed, Interrupt driven & Direct Memory Access, Synchronous & synchronous data transfer, The need for Memory Hierarchy, locality of reference principle, Memory Hierarchy, cache, main and secondary, Memory parameters, access cycle time, cost per unit. | **07L** |
| UNIT-05 | **Introduction to Parallelism:** Goals of parallelism, Instruction level parallelism, pipelining, superscaling, Processor level parallelism, Multiprocessor system overview. | **07L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Appreciate macro organization of any computing system.
CO2: Design instruction set architectures and develop their micro architectures.
CO3: Understand various digital arithmetic algorithms.
CO4: Analyze various caching and architecture memory system architectures.
CO5: Understand instruction level parallelism.

**Books and References**

1. Computer Organization and Architecture, Designing for Performance by William Stallings, Pearson Education.
2. Computer System Architecture by M. Morris Mano, Prentice Hall of India.
3. Computer Architecture: A quantitative Approach by David A. Patterson and John. L. Hennessy, Morgan Kaufmann.
4. Structured Computer Organization by Andrew S. Tanenbaum and Todd Austin, Prentice Hall of India.

| Course Name: | **Operating System** | |
|---|---|---|
| Course Code: | **CS-222** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the concepts of operating system and its management.
- To introduce the fundamental concepts scheduling of processes for a given problem instance.
- To enable the students to understand memory management techniques and implement replacement algorithms and understand and implement file systems.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Evolution of Operating Systems:** Evolution of operating systems, Types of operating systems. The process concept, system programmer's view of processes, operating system's views of processes, operating system services for process management. | **05L** |
| UNIT-02 | **CPU Scheduling:** Scheduling concepts, scheduling algorithms, algorithm evaluation, multiple processor scheduling, real time scheduling. | **06L** |
| UNIT-03 | **Concurrent Programming and Deadlocks:** Critical regions, Conditional critical regions, Monitors, Interprocess communication, Messages, Pipes, Semaphores, Modularization, Synchronization, Concurrent languages. Deadlocks: Characterization, Prevention, Avoidance, Detection and Recovery, Combined approach to Deadlock Handling, precedence graphs. | **06L** |
| UNIT-04 | **Memory Management:** Memory Management, Contiguous allocation, static-swapping, overlays, dynamic partitioned memory allocation, demand paging, page replacement, segmentation. Non-contiguous allocation, paging, Hardware support, Virtual Memory. | **07L** |
| UNIT-05 | **File Systems:** A Simple file system, General model of a file system, Symbolic file system, Access control verification, Logical file system, Physical file system, Allocation strategy module, Device strategy module, I/O initiators, Device handlers, Disk scheduling. | **06L** |
| UNIT-06 | **Networks, Security and Design Principles:** Network operating system, distributed operating system, external security, operational security, password protection, access control, security kernels, hardware security, layered approach, design principle. | **06L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:  Understand and analyze the concepts of operating system and its management.
CO2:  Illustrate the scheduling of processes for a given problem instance.
CO3:  Identify the dead lock situation and provide appropriate solution.
CO4:  Analyze memory management techniques and implement replacement algorithms.
CO5:  Understand and implement file systems.

**Books and References**

1. Operating System Concepts by J.L. Peterson and A. Silberchatz, Addison Wesley.
2. An Introduction to Operating System by Harvey M. Dietel, Addison Wesley.
3. Operating Systems - A Design Oriented Approach by C. Crowley, Irwin Publishing.
4. Operating systems by W. Stallings, Prentice Hall.
5. Modern Operating system by A.S. Tanenbaum, Prentice Hall of India.

| Course Name: | **Theory of Computation** | |
|---|---|---|
| Course Code: | **CS-223** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the basic concept of finite automata and formal languages.
- To introduce the fundamental concepts relevant to context free grammars and ability to construct grammars for specific tasks.
- To enable the students to understand the Turing Machine and push down automata.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Machines:** Basic machine, FSM, Transition graph, Transition matrix, Deterministic and non-deterministic FSMS, Equivalence of DFA and NDFA, Mealy and Moore machines, minimization of finite automata, Two-way finite automata. | **06L** |
| UNIT-02 | **Regular Sets and Regular Grammars:** Alphabet, words, Operations, Regular sets, Finite automata and regular expression, Pumping lemma and regular sets, Application of pumping lemma, closure properties of regular sets. | **05L** |
| UNIT-03 | **Formal Grammars and Languages:** Basic definitions and examples of languages, Chomsky hierarchy, Regular grammars, context free & context sensitive grammars, context free languages, non-context free languages, Chomsky normal forms, binary operations on languages. Simplification of CFG, Elimination of Useless symbols, Unit productions, Null productions, Greiback Normal form, Chomsky normal form – Problems related to CNF and GNF. | **07L** |
| UNIT-04 | **Turing Machines and Pushdown Automata:** TM model, representation and languages acceptability of TM Design of TM, Universal TM and Other modification, composite and iterated TM, Pushdown automata, Acceptance by PDA. | **05L** |
| UNIT-05 | **Computability:** Basic concepts, primitive and partial recursive function, Recursive function, Decidability, Kleene's theorem.<br><br>**Undecidibility:** Properties of recursive & recursively enumerable languages, Universal Turing machine and undecidable problem, Rice's theorem, undecidable problems. | **08L** |
| UNIT-06 | **Computational Complexity Theory:** Definition, linear speed-up, tape compression & reduction in number of tapes, Hierarchy Theorem, Relation among complexity measures, Transition lemmas and  non-deterministic hierarchies, properties of general complexity measures, the gap, speed-up, union theorem, Automatic complexity theorem. | **05L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Understand the basic concept of finite automata and formal languages.
CO2:     Demonstrate understating of regular expressions and their connection to FSM.
CO3:     Demonstrate understating of context free grammars and ability to construct grammars for specific tasks.
CO4:     Demonstrate understanding of Turing Machine and push down automata.

**Books and References**

1.    Introduction to Automata theory, Languages and Computation by John E. Hopcroft and Jeffery Ullman, Narosa Publishers.
2.    Introductory Theory of Computer Science by E.V. Krishnamurthy, East West Press.
3.    Theory of computer Science by K.L.P. Mishra, Prentice Hall of India.
4.    An Introduction to Formal Languages and Automata by Peter Linz, Jones & Bartlett Learning.
5.    Introduction to Languages and the Theory of Computation by John C. Martin, Tata McGraw Hill.

| Course Name: | **Data Structures Lab** |
|---|---|
| Course Code: | **CS-202** |

| Contact Hours/Week: **2P** | Course Credits: **01** |
|---|---|

**Course Objectives**

- To provide skills for designing & writing algorithms.
- To provide skills for writing C/C++ programs.
- To enable the students to debug programs.

**List of Experiments**

1.  Write a program to sort an array (make a dynamic array) using Bubble sort. Use 1-bit variable FLAG to signal when no interchange take place during pass. If FLAG is 0 after any pass, then list is already sorted and there is no need to continue.
2.  Write a program to search an ITEM (integer) in an array using binary search, if FOUND then delete that item from array and if NOT FOUND than insert that item in kth position (Input "k" from user).
3.  Write a program to enter records of Five students, which should contain fields like roll No., name, CGPI, semester. (a) List all record of all students having CGPI greater than k. (b) Insert a new record of student at kth position and print the final record.
4.  Implement linked list and insert and delete an element into the list.
5.  Evaluate a postfix algebraic expression with the help of stack.
6.  Implement a queue using arrays and linked list.
7.  Implement a binary tree and implement any traversal technique as you like.
8.  Implement a binary Search Tree and insert and delete a node in the BST.
9.  Implement Max Priority queue using Max Heap.
10. Implement a graph and find transpose of a graph where Transpose of a directed graph G is another directed graph on the same set of vertices with all of the edges reversed compared to the orientation of the corresponding edges in G. That is, if G contains an edge (u, v) then the converse/transpose/reverse of G contains an edge (v, u) and vice versa. Implement it with the help of adjacency list and adjacency matrix.
11. Implement Quick Sort, Merge Sort, Insertion Sort and Selection Sort.

***Note:** The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:   Design and analyze the time and space efficiency of the data structure.

CO2:   Identity the appropriate data structure for given problem.

CO3:   Apply practical knowledge on the applications of data structures.

| Course Name: | **Computer Organization and Architecture Lab** | |
|---|---|---|
| Course Code: | **CS-224** | |
| Contact Hours/Week: **2P** | | Course Credits: **01** |

**Course Objectives**

- To simulate the behavior of logic gates, adders and flip-flops.
- To understand the design of processors.
- To understand the concept of memory hierarchy.

**List of Experiments**

1. Using virtual lab, implement i) Ripple Carry Adder, ii) Carry-Look-Ahead Adder, iii) Wallace Tree Adder..
2. Using virtual lab, implement i) Synthesis of Flip Flops, ii) Registers and Counters.
3. Using virtual lab implement i) Combinational Multipliers, ii) Booth's Multiplier.
4. Using logisim simulator design i) 8 bit Arithmetic Logic Unit, ii) 16 bit Arithmetic Unit.
5. Simulate Datapath for processor using procsim simulator.
6. Design control path using Finite State Machine for CPU.
7. Write a program to understand register and memory allocation in registers for MIPS architecture.
8. Design Memory (Design of a RAM Cell and 4*4 RAM).
9. Write a program to find the factorial of a number in MIPS, C, FORTRAN and Python. Check the maximum data size for which factorial could be calculated. Find out how the overflow conditions are handled by different processors.
10. Write a program to implement Karnaugh Map
11. Implement the cache memory approaches i) Associative cache Design, ii) Direct Mapped cache Design using ParaCache simulator.

*Note: The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Learn various computer organization and architecture concepts.
CO2: Apply various arithmetic operations and processor design methods.
CO3: Analyze the architecture and functionality of central processing unit and memory hierarchy.

| Course Name: | **Operating System Lab** |
| Course Code: | **CS-225** |

| Contact Hours/Week: **2P** | Course Credits: **01** |

**Course Objectives**

- To enlighten the student with knowledge base in compiler design and its applications.
- To implement Lexical Analyzer using Lex tool and Syntax Analyzer or parser using YACC Tool.
- To implement front end of the compiler by means of generating Intermediate codes.
- To implement code optimization techniques.

**List of Experiments**

1. Practice various commands/utilities in MS Windows and Linux OS for the following:
   - File management: create, copy, delete, link, display content, printing
   - Directory handling: create, delete, relocate, display content, sharing
   - Security features: setting/resetting file permissions to read, write, execute
   - Network configuration
2. Learn the top command to display resource utilization statistics of processes.
3. Simulate various CPU scheduling algorithms (FCFS, SJF-preemptive & no preemptive, Priority Scheduling-preemptive & no preemptive and Round Robin). Run your simulator for each scheduling policy, with a variety of quantum values. For each version of the simulator, for each input data file, for each quantum value, plot the completion time, throughput, average job elapsed time and average job waiting time. Analyze the behavior of your scheduler.
4. Write a program to simulate Banker's algorithm for deadlock handling.
5. Simulate memory allocation algorithms (First Fit and Best Fit).
6. Simulate different page replacement strategies (FIFO, LRU and Optimal) and compare their performance.
7. Using the fork(…) system call, write a program in which the parent creates a child and the child creates a grandchild. Synchronize the processes so that the grandchild first displays a message (and then terminates), the child displays a message (and then terminates), and finally the parent displays a message.
8. Simulate (at least three) disk scheduling protocols.
9. Write a program to display the IP address and MAC address of a machine.
10. Write script to implement background process that will continually print current time in upper right corner of the screen, while user can do his/her normal job at prompt.
11. Write shell script to show various system configuration like
    - Currently logged user and his logname
    - Your current shell
    - Your home directory
    - Your current path setting
    - Your current working directory
    - Show Currently logged number of users
    - About your OS and version, release number, kernel version
    - Show all available shells
    - Show computer CPU information like processor type, speed, etc.
    - Show memory information
    - Show hard disk information like size of hard-disk, cache memory, model, etc.
    - File system (Mounted)
12. Installation and configuration of MS Windows and Linux OS.


*Note:* The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Implement elementary UNIX system commands.
CO2: Devise programs to test synchronization problems.
CO3: Design and develop user level thread library.
CO4: Design and implement file system.

| Course Name: | **Analysis and Design of Algorithms** | |
|---|---|---|
| Course Code: | **CS-311** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the asymptotic notations to analyze the performance of algorithms.
- To introduce the fundamental concepts various problem solving techniques such as divide and conquer, greedy algorithm, etc.
- To enable the students to understand the concepts of P, NP, NP-hard and NP-complete problems.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Algorithms Introduction:** Algorithm Design paradigms- motivation, concept of algorithmic efficiency, run time analysis of algorithms, Asymptotic Notations. | **03L** |
| UNIT-02 | **Divide and Conquer Approach:** Structure of divide-and-conquer algorithms: sets and disjoint sets: Union and Find algorithms, quick sort, Finding the maximum and minimum, Quick Sort, Merge sort, Heap and heap sort. | **05L** |
| UNIT-03 | **Greedy Algorithms:** Optimal storage on tapes, Knapsack problem, Job sequencing with deadlines, Minimum Spanning trees: Prim's algorithm and Kruskal's algorithm, Huffman codes. | **05L** |
| UNIT-04 | **Graph Algorithms:** Representation of graphs, BFS, DFS, Topological sort, strongly connected components; single source shortest paths: Bellmen-Ford algorithm, Dijkstra's algorithm; All pairs shortest path: The Warshall's algorithm. | **06L** |
| UNIT-05 | **Dynamic Programming:** Overview, difference between dynamic programming and divide and conquer, Matrix chain multiplication, Traveling salesman Problem, longest Common sequence, 0/1 knapsack.<br>**Backtracking:** 8-Queen Problem, Sum of subsets, graph coloring, Hamiltonian cycles. | **09L** |
| UNIT-06 | **Branch and Bound:** LC searching Bounding, FIFO branch and bound, LC branch and bound application: 0/1 Knapsack problem, Traveling Salesman Problem.<br>**Computational Complexity:** Complexity measures, Polynomial vs. nonpolynomial time complexity; NP-hard and NP-complete classes, examples. | **08L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Understand asymptotic notations to analyze the performance of algorithms.

CO2:     Understand and apply various problem solving techniques

CO3:     Solve given problem by selecting the appropriate algorithm design technique and justify the selection.

CO4:     Know the concepts of P, NP, NP-hard and NP-complete problems.

**Books and References**

1.   Fundamentals of Computer Algorithms by E. Horowitz and S. Sahni, Galgotia.
2.   Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, MIT Press, Cambridge.
3.   The Design and Analysis of Computer Algorithms by A.V. Aho, J.E. Hopcroft and J.D. Ullman, Addison Wesley.

| Course Name: | **Data Base Management Systems** | |
|---|---|---|
| Course Code: | **CS-312** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the DBMS concept, data models and architecture.
- To introduce the fundamental concepts relevant to ER model and its mapping to relational model, relational algebra and SQL.
- To enable the students to understand the factors that causes the concurrency and recovery strategies for DBMS.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Basic Concepts:** Introduction to File and Database systems- Database system structure – concepts and architecture, date models, schemas & instances, DBMS architecture & data independence, database languages & interfaces, Data Model, ER model. | **04L** |
| UNIT-02 | **Relational Models:** SQL – Data definition- Queries in SQL-relational model concepts, relational model constraints, relational algebra, SQL- a relational database language: date definition in SQL, view and queries in SQL, specifying constraints and indexes in SQL; relational database management systems-Updates, Views, Integrity and Security, Relational Database design, Functional dependences and Normalization for Relational Databases, normal forms based on primary keys, (1NF, 2NF, 3NF & BCNF), lossless join and dependency preserving decomposition, converting ER-diagrams into relations**.** | **09L** |
| UNIT-03 | **Data Storage and query Processing:** Record storage and Primary file organization- Secondary storage Devices, Operations on Files, Heap File, Sorted Files, Hashing Techniques, Index Structure for files, Different types of Indexes- B-Tree - B+Tree, Query Processing. | **09L** |
| UNIT-04 | **Transaction Management:** Transaction Processing, Need for Concurrency control, Desirable properties of Transaction, Schedule and Recoverability, Serializability and Schedules; Concurrency Control, Types of Locks, Two Phases locking, Deadlock, Time stamp based concurrency control , Recovery Techniques, Concepts- Immediate Update- Deferred Update, Shadow Paging. | **09L** |
| UNIT-05 | **Current Trends:** Introduction to Distributed and parallel databases, Deductive Databases, Multimedia Databases, Real-Time Databases. | **05L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Understand basic concepts and appreciate the applications of database systems.
CO2: Explain relational data model, entity-relationship model, relational database design, relational algebra and SQL.
CO3: Understand basic database storage structures and access techniques:
CO4: Improve the database design by normalization.
CO5: Understand concurrency and recovery strategies for DBMS.

**Books and References**

1. Database System Concepts by Abraham Silberschatz, Henry F. Korth and S. Sudarshan, McGraw-Hill.
2. Fundamental Database Systems by Ramez Elmasri and Shamkant B. Navathe, Pearson Education.
3. An Introduction to Database Systems by C.J.Date, Addison Wesley.
4. Fundamentals of Database Systems by Ramez Elmasri  and Shamkant B. Navathe, Addison Wesley.
5. Database System Implementation by Hector Garcia–Molina, Jeffrey D.Ullman and Jennifer Widom, Pearson Education.
6. Database System, Design, Implementation and Management by Peter Rob and Corlos Coronel, Thompson Learning Course Technology.

| Course Name: | **Compiler Design** | |
|---|---|---|
| Course Code: | **CS-313** | |
| Course Type: | **Core** | |
| Contact Hours/Week:  **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the different phases of compilation process
- To introduce the fundamental concepts relevant to compiler design techniques.
- To enable the students to understand the code optimization and code generation.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction to Compilers:** Compiler and translators need of translators, structure of a compiler, lexical analysis, syntax Analysis. Basic Parsing Techniques: Parsers shift-reduce parsing, predictive parsing. | **04L** |
| UNIT-02 | **Automatic Construction of Efficient Parsers:** LR parsers, canonical collection of LR(0) items, construction canonical LR parsing tables, construction LALR and SLR parsing tables using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables, construction LALR sets of items. | **10L** |
| UNIT-03 | **Syntax-Directed Translation:** Syntax directed translation schemes, implementation of syntax directed translation, intermediate code, postfix notation parse trees and syntax trees, three address code, quadruples and triples, translation of assignment statements, postfix translation with top down parser**.** | **08L** |
| UNIT-04 | **Symbol Tables:** Contents of a table, data structures for symbol tables, representing scope information. Error detection and recovery: Errors, lexical-phase errors, syntax-phase errors, semantic errors. | **07L** |
| UNIT-05 | **Introduction to Code optimization:** The principal source of optimization, loop optimization, DAG representation of basic blocks, value numbers and algebraic laws, global data flow analysis. Code Generation: Object programs, problems in code generation, machine model, simple code generator, register allocation and assignment, code generation from DAG's, peephole optimization. | **07L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Understand different phases of compilation process.

CO2:     Analyze and implement various parsing techniques.

CO3:     Understand and analyze intermediate code.

CO4:     Realize the importance of code optimization and code generation.

**Books and References**

1.   Compilers Principles, Techniques and Tools by Alfred V. Aho, Monica S. Lam, Ravi Sethi and Jeffrey D Ullman, Pearson Education.
2.   Compiler Design in C by Allen I. Holub, Prentice Hall of India.
3.   Introduction to Compiler Techniques by J.P. Bennet, Tata McGraw-Hill.
4.   Practice and Principles of Compiler Building with C by Henk Alblas and Albert Nymeyer, Prentice Hall of India.
5.   Lex & Yacc by John R Levine, Tony Mason and Doug Brown, O'reilly.
6.   Engineering a Compiler by Keith D. Cooper and  Linda Torczon, Morgan Kaufmann, Elsevier.

| Course Name: | **Computer Graphics** | |
|---|---|---|
| Course Code: | **CS-314** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about the computer graphics display technologies.
- To introduce the fundamental concepts relevant to various 2D transformations, clipping algorithms to 2D primitives and demonstrate the 3D transformation concepts to model an object.
- To enable the students to understand the implementation of various rendering algorithms.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction to Computer Graphics:** Overview of Graphics Systems, Display Devices, Hard copy Devices. Interactive Input Devices, Display Processors, The Graphical Kernel System, Output Primitives, Graphics hardware: Display technology, random scan, raster scan display processing, graphics software and standards. | **05L** |
| UNIT-02 | **Raster Scan Graphics**: Points, lines, circles and ellipses as primitives, scan conversion algorithms for primitives, Fill area primitives including scan-line polygon filling, inside-outside test, boundary and flood-fill, character generation, line attributes, area-fill attributes, character attributers, antialiasing. | **07L** |
| UNIT-03 | **Two-Dimensional Geometric Transformation & Viewing:** Basic Transformation, Translation, Rotation, Scaling, Other Transformation Reflection, Shear, Transformation functions, Window to viewport co-ordinate transformation, Clipping Operations, Point Clipping, Line Clipping, Polygon Clipping. | **07L** |
| UNIT-04 | **Three- Dimensional Concepts & Object Representations:** Three Dimensional Display Methods, Parallel Projection, Perspective Projection, Translation, Rotation, Scaling, Composite Transformation, Three dimensional Transformation function, Polygon Surfaces, Curved Lines and surfaces, Bezier Curves and surfaces, B-Spline Curves and surfaces | **08L** |
| UNIT-05 | **Visible Lines and Visible Surfaces:** Visual Realism, Hidden line and hidden surface removal: depth buffer algorithm, geometric computations, scan line coherence algorithms, area coherence algorithms, priority algorithm, visible surface detection concepts, back-face detection, depth buffer method,  light sources, illumination methods (ambient, diffuse reflection, specular reflection), shading, Color models: properties of light, XYZ, RGB, YIQ and CMY color models, modeling methods, Transparency, Refraction effects in transparent materials, Simple Transparency Models, Shadows, Texture. | **09L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Understand the computer graphics display technologies.

CO2: Implement various 2D transformations.

CO3: Apply the clipping algorithms to 2D primitives.

CO4: Demonstrate the 3D transformation concepts to model an object.

CO5: Understand and implement various rendering algorithms.

**Books and References**

1. Procedural Elements for Computer Graphics by D.F. Rogers, McGraw Hill.
2. Mathematical Elements for Computer Graphics by D.F. Rogers, McGraw Hill.
3. Computer Graphics, with OpenGL by Donald Hearn, M. Pauline Baker, Pearson Education.
4. Computer Graphics by John F Hughes, Andries Van Dam, Morgan McGuire, David F Sklar, James D Foley, Steven K Feiner and Kurt Akeley, Person Education

| Course Name: | **Data Base Management Systems Lab** |
|---|---|
| Course Code: | **CS-315** |

| Contact Hours/Week: **2P** | Course Credits: **01** |
|---|---|

**Course Objectives**

- To impart knowledge about the DBMS concept, data models and architecture.
- To introduce the fundamental concepts relevant to ER model and its mapping to relational model, relational algebra and SQL.
- To enable the students to understand the factors that causes the concurrency and recovery strategies for DBMS.

**List of Experiments**

Gather the Required Information, Draw ER Diagrams, Map ER Diagrams to Tables, Perform Normalization, Write Procedures, Execute Queries, Create User Interfaces and Generate Reports:

1. Online Examination System
2. Development of Online Course Portal for the Campus
3. Student Attendance Management System
4. Student Information System
5. Student Result Alert System
6. Time Table Management System
7. Examination Date Sheet Management System
8. Supplementary Examination Registration System
9. Training and Placement Cell Management
10. Hostel Allotment System
11. Campus Network Complaint System
12. Hospital Management System
13. Civil/Electrical Works Repair Information System

*Note: The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list. The lab work is based on the large scale project undertaken from the tentative lit provided or suggested by the concerned teacher.*

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Understand basic concepts and develop application using DBMS tools and techniques.

CO2:    Use  relational data model, entity-relationship model, relational database design, relational algebra and SQL

CO3:    Improve the database design by normalization.

---

| Course Name: | **Compiler Design Lab** | |
|---|---|---|
| Course Code: | **CS-316** | |
| Contact Hours/Week: **2P** | | Course Credits: **01** |

**Course Objectives**

- To enlighten the student with knowledge base in compiler design and its applications
- To implement Lexical Analyzer using Lex tool & Syntax Analyzer or parser using YACC Tool
- To implement front end of the compiler by means of generating Intermediate codes.
- To implement code optimization techniques

**List of Experiments**

1. Write a C/C++ program to search the number of occurrences of any word, string and sub sequence in a given string
2. Write a C/C++ program to find the number of occurrences of any word in a file and replace that with another word.
3. Implement the lexical analyzer using JLex, Flex or other lexical analyzer generating tools and write the following program using LEX.
   - Program to count the number of characters, words, spaces and lines in a given input file.
   - Program to count the number of comment lines in a given C program. Also eliminate them and copy the resulting program into separate file.
   - Program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately.
   - Program to recognize and count the number of identifiers in a given input file.
4. Write a program to check whether a string belongs to the grammar or not. Write a program to generate a parse tree.
5. Write a program to check whether a grammar is left recursive and left factoring. Write a program to remove left recursion and left factoring.
6. Write a program to implementation FIRST and FOLLOW for the given grammar.
7. Write a program to implementation to implement recursive-Descent and Non-recursive Predictive parser.
8. Write a program to check whether a given grammar is LL (1) or not.
9. Write a C/C+ + program to implementation of shift reduce parsing Algorithm
10. Write a C/C++ program to construct i) LR Parsing table and Implement LR parsing algorithm, ii) CLR Parsing table and implement CLR parsing algorithm, iii) LALR Parsing table and implement LALR parsing algorithm.
11. YACC program to recognize i) a valid arithmetic expression that uses operators +, -,* and /, ii) strings 'aabb','abbb', 'ab' and 'a' using the grammar ($a^n b^n, n >= 10$), where a and b are input symbols of the grammar.
12. Write a C program to implement i) the syntax-directed definition of "if E then S1" and "if E then S1 else S2", ii) a Symbol Table with function to create, insert, modify, search and display.

*Note: The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Design Lexical analyzer for given language using C and LEX tools.

CO2:     Design and convert BNF rules into YACC form to generate various parsers.

CO3:     Generate machine code from the intermediate code forms.

CO4:     Implement Symbol table and compiler design aspects.

---

| Course Name: | **Computer Graphics Lab** | |
|---|---|---|
| Course Code: | **CS-317** | |
| Contact Hours/Week: **2P** | | Course Credits: **01** |

**Course Objectives**

- Understand the need of developing graphics application.
- Learn algorithmic development of graphics primitives like: line, circle, polygon, etc.
- Learn the representation and 2D/ 3D transformation of graphical images and pictures.

**List of Experiments**

1. Program to implement DDA algorithms for line.
2. Program to implement Bresenham's algorithms for line, circle and ellipse drawing.
3. Program to implement Mid Point Circle algorithm.
4. Program to implement Mid Point Ellipse algorithm.
5. Program to perform 2D Transformations such as translation, rotation, scaling, reflection and sharing.
6. Program to implement Cohen–Sutherland 2D clipping and window–viewport mapping.
7. Program to implement Liang Barsky Line Clipping Algorithm.
8. Program to perform 3D Transformations such as translation, rotation and scaling.
9. Program for conversion between color models.
10. Program to perform animation using any Animation software
11. Program to perform basic operations on image using any image editing software
12. Program to draw different shapes such as hut, face, kite, fish etc.

*Note: The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Draw Geometric primitives using programs.

CO2: Implement representation & transformations on objects using program.

CO3: Use clipping and editing algorithm on images.

| Course Name: | **Distributed Systems** | |
|---|---|---|
| Course Code: | **CS-321** | |
| Course Type: | **Core** | |
| Contact Hours/Week:  **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the concepts of various architectural and fundamental models in distributed systems.
- To introduce the fundamental concepts relevant to client server and peer to peer interactions between clients and sever.
- To enable the students to understand the factors that cause the intricacies involved in distributed transactions.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** Characterization of Distributed System, Examples of distributed systems, Resource sharing and the World Wide Web, WWW 1.0, 2.0, 3.0, Challenges. System Models: Architectural models, S/w layers, system architecture and variants, Interface and Objects, Design requirements for distributed architectures, Fundamental Models: Interaction Model, Failure Model and Security Model. | **06L** |
| UNIT-02 | **Interprocess Communication:** The API for the Internet protocols, characteristics of interprocess communication, Sockets, UDP Datagram communication, TCP stream communication, External data representation and marshalling, Client-server communication, Group communication, IP multicast, Network virtualization, Overlay networks. | **07L** |
| UNIT-03 | **Distributed Objects and Remote Invocation:** Indirect Communication, Operating System Support, Distributed File Systems, File service architecture, Andrew File system. File System: Features-File model, File accessing models, File sharing semantics naming: Identifiers, Addresses, Name Resolution, Name Space Implementation, Name Caches, LDAP. | **08L** |
| UNIT-04 | **Time and Global States:** Clocks, events and process states, Skew between computer clocks in a distributed system, Coordinated Universal Time (UTC),  physical time and clocks, Clock correctness, Cristian's method for an asynchronous system, Berkeley algorithm, NTP,  logical time and clocks, vector clocks, global states, distributed garbage collection. Distributed Debugging, Global states and consistent cuts, Snapshot Algorithm. | **08L** |
| UNIT-05 | **Coordination and Agreement:** Distributed mutual exclusion, elections, multicast communication, coordination agreement, consensus and related problems. Transactions and Concurrency Control, Transactions, Nested transactions, Locks, Optimistic concurrency control, Timestamp ordering, Atomic Commit protocols, Distributed deadlocks,  Replication. | **07L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Understand concepts of various architectural and fundamental models in distributed systems.
CO2:    Implement client server and peer to peer interactions between clients and sever.
CO3:    Analyze time synchronization protocols and select a suitable time synchronization protocol.
CO4:    Analyze and evaluate various election and agreement protocols in distributed environment.
CO5:    Analyze various intricacies involved in distributed transactions.

**Books and References**

1.    Distributed Systems: Concepts and Design by G. Coulouris, J. Dollimore, and T. Kindberg, Pearson Education.
2.    Distributed Systems: Principles and Paradigms by A.Tanenbaum and Maarten van Steen, Prentice Hall of India.
3.    Advanced Concepts in Operating Systems by M. Singhal and  N. Shivaratri, Tata McGraw Hill.
4.    Distributed Systems - An Algorithmic Approach by Sukumar Ghosh, Chapman and Hall/CRC.
5.    Distributed Algorithms: Principles, Algorithms, and Systems by D. Kshemkalyani and M. Singhal, Cambridge University Press.

| Course Name: | **Software Engineering** | |
|---|---|---|
| Course Code: | **CS-322** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the concept of software development and software engineering
- To introduce the fundamental concepts relevant to comprehend different software engineering process models.
- To enable the students to understand the factors those affects the design of software projects and do the cost estimation.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** Problem domain, software engineering challenges, software engineering approach. Software Processes: Software process, characteristics of software process, software development process models, other processes. | **04L** |
| UNIT-02 | **Software Requirements Analysis and Specification:** Software requirements, problem analysis, requirements specification, functional specification with use cases, validation, matrices. | **04L** |
| UNIT-03 | **Software Architecture:** Role of software architect, architecture views, component and connector view, architecture style for C & C view, discussion and evaluating architectures. Planning a software project: Effort estimation, project scheduling and staffing, software configuration management plan, quality assurance plan, risk management, project monitoring plan. | **06L** |
| UNIT-04 | **Function Oriented Design:** Design principles, module level concepts, design notation and specification, structured design methodology, verification, metrics. Object oriented design: OO concepts, design concept, Unified Modeling Language, design methodology, metrics. | **05L** |
| UNIT-05 | **Detailed Design, Software Measurements, Metrics and Models:** Detailed design and PDL, verification, Metrics and their scope, Qualities of a good Software metrics, classification of metrics, Cost estimation models COCOMO, Quality attributes, SQA, Quality Standards, ISO 9000 and CMM. | **06L** |
| UNIT-06 | **Coding:** Programming principles and guidelines, coding process, refactoring, verification, metrics. Testing: Testing fundamentals, black-box testing, white-box testing, testing process, defect analysis and prevention, metrics - reliability estimation. CASE Tools: Types of CASE tools, advantages and components of CASE tools, Unified Modeling Language (UML). | **06L** |
| UNIT-07 | **IPR and Copyright:** Introduction and the need for intellectual property right (IPR), IPR in India – Genesis and Development, Patents: Searching, Drafting, Filling and Granting of Patents; Copyrights: Coverage and Need; Trademarks: Types, Rights, Kinds of signs of Trademarks; Geographical Indications. | **05L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Understand and analyze the concept of software development and software engineering.
CO2: Compare and comprehend different software engineering process models.
CO3: Design of software projects and do the cost estimation.
CO4: Apply different software testing techniques.

**Books and References**

1. An Integrated Approach to Software Engineering by Pankaj Jalote, Narosa Publishing.
2. Software Engineering: A Practitioner's Approach by Roger. S. Pressman, Tata McGraw Hill.
3. Fundamentals of Software Engineering by Rajib Mall, Prentice Hall of India.
4. UML Bible by Tom Pender, Wiley Dreamtech.
5. Software Engineering by Ian Sommerville, Addison-Wesley

| Course Name: | **Digital Image Processing** | |
|---|---|---|
| Course Code: | **CS-323** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L + 1T** | | Course Credits: **04** |

**Course Objectives**

- To impart knowledge about the fundamental steps of digital image processing.
- To introduce the fundamental concepts relevant to digital image processing.
- To able the students to understand the various image segmentation techniques and morphological algorithms.
- To able the students to understand the various image compression techniques.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** Digital image representation, Fundamental steps in image processing, Elements of Digital Image processing systems, Elements of visual perception, Image model, Sampling and quantization, Relationship between pixels, Imaging geometry. | **06L** |
| UNIT-02 | **Image Enhancement:** Enhancement by point processing, Intensity transformation, Histogram processing, Image subtraction, Image averaging, Spatial filtering, Smoothing filters, Sharpening filters, Frequency domain: Low-Pass, High-Pass, Homomorphic filtering. | **06L** |
| UNIT-03 | **Image Segmentation:** Detection of discontinuities, Edge linking and boundary detection, Thresholding, Region oriented segmentation, Use of motion in segmentation, Spatial techniques, Frequency domain techniques. | **06L** |
| UNIT-04 | **Image Restoration:** A model of the image degradation/restoration process, noise models, restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the Fourier transform and the frequency domain, estimating the degradation function. Color Image Processing. | **06L** |
| UNIT-05 | **Morphological Image Processing:** Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphological algorithms. | **06L** |
| UNIT-06 | **Image Compression:** Coding redundancy, Inter-pixel redundancy, fidelity criteria, image compression models, error-free compression, lossy predictive coding, image compression standards. | **06L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Understand fundamental steps of digital image processing.
CO2:     Understand and implement image enhancement techniques.
CO3:     Implement and compare various image compression techniques.
CO4:     Understand and implement pattern recognition and classification techniques.

**Books and References**

1.     Digital Image Processing, by  R. Gonzalez and R. E. Wood, Prentice Hall of India.
2.     Digital Image Processing using MATLAB by R. Gonzalez, R. E. Wood and Steven L. Eddins Gatesmark Publishing.
3.     Introductory Computer Vision and Image Processing by   Andrian Low. McGraw Hill.
4.     Pattern Recognition-Statistical, Structural and neural approach by  Robert Schalkoff, John Willey & Sons.
5.     Fundamentals of Digital Image Processing by A K Jain, Prentice Hall.
6.     Digital Image Processing by W.K. Pratt, McGraw Hill.

| Course Name: | **Computer Networks** | |
|---|---|---|
| Course Code: | **CS-324** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about the network models and architectures.
- To introduce the fundamental concepts relevant to performance of various routing protocols and design of new routing protocol.
- To impart knowledge on designing and building a complete networking solution as per the requirement of an organization.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introductory Concepts:** Goals and Applications of Networks, LAN, WAN, MAN, Wireless network, Network software: Protocol hierarchies, design issues of layers, Interfaces and services. Reference Model: The OSI reference model, TCP/IP reference model, Example networks: Novell Netware, The ARPANET, The Internet, X-25 Networks, network standards. | **05L** |
| UNIT-02 | **Physical Layer:** Fourier analysis, Maximum data rate of a channel, Transmission media, Wireless transmission, Narrowband ISDN, Broadband ISDN and ATM, Virtual circuits, Circuit switching, Communication satellite. | **04L** |
| UNIT-03 | **Data Link Layer:** Data link layer design issues, services provided to network layers, Framing, Error control, Flow control, Error detection and correction, Elementary data link protocols, An unrestricted Simplex protocol, A Simplex Stop-and-Wait protocol, Simplex Protocol for a noisy channel, Sliding Window protocols, A one-bit Sliding protocol, A protocol using go-back-N, A protocol using selective repeat, Protocol specification and verification, Example data link protocol-HDLC, PPP and SLIP. | **06L** |
| UNIT-04 | **Medium Access Sublayer:** Channel Allocations, Static and dynamic allocation in LAN and MAN, Multiple Access protocols, ALOHA, Carrier Sense multiple access protocols, WDMA protocols, Wireless protocols, Collision free protocols, Limited contention protocols, IEEE standard 802.3 and Ethernet, IEEE standard 802.4, Token bus IEEE standard 802.5, Token Ring, Distributed Queue Dual bus, Logical link control, bridges, High speed LAN, Satellite networks. | **06L** |
| UNIT-05 | **Network Layer:** Network Layer design issue, Routing algorithms, Congestion Control Algorithms, Internetworking.<br><br>**Transport Layer:** Transport services, Design issues, elements of transport protocols, simple transport protocols, Connection management, TCP, UDP. | **10L** |
| UNIT-06 | **Session, Presentation and Application Layer:** Session Layer, Design issues, remote procedure call. Presentation Layer, Design issues, Data compression techniques, cryptography. Application Layer - File Transfer, Access and Management, Electronic mail, Virtual Terminals, Other applications, Example Networks - Internet and Public Networks. | **05L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Understand network models and architectures.

CO2: Identify the pros and cons of choosing a suitable MAC layer protocol.

CO3: Analyze the performance of various routing protocols and design of new routing protocol.

CO4: Solve basic network design problems using knowledge of common local and wide area network architectures.

**Books and References**

1. Computer Networks by A.S. Tanenbaum, Prentice Hall of India.
2. Computer Networking: A Top-Down Approach Featuring the Internet by J. Kurose and K.W. Ross, Addison-Wesley.
3. Data and Computer Communication by W. Stallings, Prentice Hall of India.

| Course Name: | **Digital Image Processing Lab** | |
|---|---|---|
| Course Code: | **CS-325** | |
| Contact Hours/Week: **2P** | | Course Credits: **01** |

**Course Objectives**

- Describe and explain basic principles of digital image processing.
- Design and implement algorithms that perform basic image processing (e.g. noise removal and image enhancement).
- Design and implement algorithms for advanced image analysis (e.g. image compression, image segmentation).
- Assess the performance of image processing algorithms and systems.

**List of Experiments**

1. Implement the spatial image enhancement functions on a bitmap image –Mirroring (Inversion)
2. Implement the spatial image enhancement functions on a bitmap image –Rotation (Clockwise)
3. Implement the spatial image enhancement functions on a bitmap image –Enlargement (Double Size)
4. Implement   i) Low Pass Filter, ii) High Pass Filter
5. Implement  i) Arithmetic Mean Filter, ii) Geometric Mean Filter
6. Implement Smoothing and Sharpening of an eight bit color image
7. Implement  i) Boundary Extraction Algorithm, ii) Graham's Scan Algorithm
8. Implement i)  Edge Detection, ii) Line Detection
9. Implement Image segmentation using OSTU and Thresholding Method
10. Implement data compression using JPEG algorithm.

*Note: The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Analyze general terminology of digital image processing.

CO2:    Examine various types of images, intensity transformations and spatial filtering.

CO3:    Develop Fourier transform for image processing in frequency domain.

CO4:    Implement image process and analysis algorithms.

CO5:    Apply image processing algorithms in practical applications.

| Course Name: | **Computer Networks Lab** | |
|---|---|---|
| Course Code: | **CS-326** | |
| Contact Hours/Week: **2P** | | Course Credits: **01** |

**Course Objectives**

- To create client and server applications using the Sockets API and the implementation of Data link layer protocol and TCP layer
- To conduct computer communication network simulations.
- To have a hands on experience of computer network simulation and modeling techniques using various simulation software.

**List of Experiments**

1. To become familiar with networking accessories and facilities in the Department of Computer Science and Engineering:
   - Find out what networking devices are installed in the department
   - Describe the network type and topology of the department
   - File and printer sharing in different OSs
   - Network address configuration in different OSs
   - Finding IP and MAC address in different OSs
   - Workgroup and domain configuration
   - Use of utilities: arp, ipconfig/ifconfig, tracert, nslookup
2. Examine packets flow across a network segment and see the operation of various Internet protocols across the different layers in TCP/IP stack. (Hint: Use utilities netstat, snoop, tcpdump, …)
3. Use UNIX sockets to implement a simple client and server that communicate over the network. (Reference: Unix Network Programming by W. Richard Stevens, Prentice Hall)
4. Simulate various multiple access protocols (Aloha, slotted Aloha, p-persistent and non-persistent) and compare their performance at different loads.
5. Write a program to display the IP address and MAC address of a machine.
6. Implement ARP and RARP protocols for CSE LAN.
7. Install Ethereal on a computer. Set Ethereal to capture with a filter option of your choice. Load a webpage or send an email to a friend and stop capturing. Analyze the packets. See if you can read any or all of the data transmitted. Write down your findings.
8. Write a program to simulate routing using flooding. Each packet should contain a counter that is decremented on each hop. When the counter gets to zero, the packet is discarded. Time is discrete, with each line handling one packet per time interval. Make three versions of the program: all lines are flooded, all lines except the input line are flooded, and only the (statically chosen) best k lines are flooded. Compare flooding with deterministic routing (k = 1) in terms of both delay and the bandwidth used.
9. Write a program that simulates a computer network using discrete time. The first packet on each router queue makes one hop per time interval. Each router has only a finite number of buffers. If a packet arrives and there is no room for it, it is discarded and not transmitted. Instead, there is an end-to-end protocol, complete with timeouts and acknowledgement a packet, which eventually regenerates the packet from the source router. Plot the throughput of the network as a function of the end-to-end timeout interval, parameterized by error rate.
10. Design and implement a chat system that allows multiple groups of users to chat. A chat coordinator resides at a well-known network address, uses UDP for communication with chat clients, sets up chat servers for each chat session, and maintains a chat session directory. There is one chat server per chat session. A chat server uses TCP for communication with clients. A chat client allows users to start, join, and leave a chat session. Design and implement coordinator, server, and client code.

*Note: The concerned Course Coordinator will prepare the actual list of experiments/problems at the start of semester based on above generic list.*

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Ability to invoke analytical studies of Computer Networks through network simulation.

CO2: Technical knowhow of the various components in simulation toolkits and their importance in designing a real network.

| Course Name: | **Engineering Economics and Accountancy** | |
|---|---|---|
| Course Code: | **HS-404** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about the Economics and its applicability to the Engineers
- To introduce the fundamental concepts of economics
- To enable the students to understand the factors that causes the changes in economic conditions of the entrepreneur

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction to Engineering Economics:** Definitions, Nature, Scope and application; Difference between Micro Economics and Macro Economics; Theory of Demand & Supply: Meaning, Determinants, Law of Demand, Elasticity of demand, Demand Forecasting, Law of Supply, Equilibrium between Demand & Supply. | **06L** |
| UNIT-02 | **Production and Cost**: Production functions, Isoquant, Least Cost combination, Laws of Returns to Scale. Economics and Diseconomies of Scale of production, Cost and Cost curves, Revenue and Revenue curve, Break even analysis. | **06L** |
| UNIT-03 | **Costing and Appraisal:** Cost elements, Economic cost, Accounting cost, Standard cost, Actual cost, Overhead cost, Cost control, Criteria of project appraisal, Social cost benefit analysis | **05L** |
| UNIT-04 | **Markets**: Meaning, Types of Markets, Characteristics (Perfect Competition, Monopoly, Monopolistic Competition, Oligopoly) Price and Output Determination; Product Differentiation; Selling Costs; Excess Capacity. | **05L** |
| UNIT-05 | **Money**: Meaning, Functions, Types; Monetary Policy- Meaning, Objectives, Tools; Fiscal Policy:-Meaning, Objectives, Tools.<br>**Banking:** Meaning, Types, Functions, Central Bank: its Functions, concepts CRR, Bank Rate, Repo Rate, Reverse Repo Rate, SLR. | **04L** |
| UNIT-06 | **Depreciation:** Meaning of depreciation, causes, object of providing depreciation, factors affecting depreciation, Methods of Depreciation: Straight line method, Diminishing balance method, Annuity method and Sinking Fund method | **04L** |
| UNIT-07 | **Financial Accounting:** Double entry system (concept only), Rules of Double entry system, Journal(Sub-division of Journal) , Ledger, Trial Balance  Preparation of final accounts-Trading Account. Profit and Loss account, Balance Sheet. | **06L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Identify the challenges of the economy as entrepreneur/manufacturer as well as consumer

CO2:     Describe the economic system at the micro and macro level

CO3:     Apply principles of economics and accountancy in the professional, personal and societal life

CO4:     Assess the role of engineering economics and accounting in attaining economic efficiency

**Books and References**

1. Principles of Micro Economics by William A. Mceachern and Simrit Kaur, Cengage Publication.
2. Managerial Economics by Craig Peterson and W Cris Lewis, PHI Publication.
3. Modern Microeconomics by A. Koutsoyiannis, Macmillan.
4. Managerial Economics Theory and Applications by D. M.Mithani. Himalaya   Publication House.
5. Fundamental of Managerial Economics Mark Hirschey, South Western Educational Publishing.
6. Engineering Economics by E. Paul DeGarmo, Prentice Hall.
7. Financial Accounting–A Managerial Perspective by R. Narayanaswamy, PHI.
8. Introduction to Accounting by J.R. Edwards and Howard J Mellett, Sage Publication.
9. Cost Accounting by Jawahar Lal, Tata McGraw Hill.
10. Project Planning Analysis, Selection, Implementation and Review by Prasanna Chandra, Tata McGraw Hill.

| Course Name: | **Artificial Intelligence** | |
|---|---|---|
| Course Code: | **CS-411** | |
| Course Type: | **Core** | |
| Contact Hours/Week:  **3L** | | Course Credits: **03** |

**Course Objectives**

- To learn the concepts of Artificial Intelligence.
- To learn the methods of solving problems using Artificial Intelligence.
- To introduce the concepts of Expert Systems and machine learning.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Fundamental Concepts:** Introduction to AI, Brief history, AI Applications, NLP, vision, robotics, etc. AI techniques, Introduction to intelligent agent, Agents and rationality, task environments, agent architecture types, Artificial Intelligence programming techniques. | **04L** |
| UNIT-02 | **Problems Solving:** Problem as a state –space, Solving problems by Searching, DFS, BFS, UCS, choice of a search algorithm, heuristic search techniques, constraint satisfaction problems, stochastic search methods, The A* algorithm and its properties, evolutionary search algorithms, genetic algorithm, Particle Swarm Optimization. | **06L** |
| UNIT-03 | **Game Playing:** Algorithms, minimax and its shortcomings, improving minimax using alpha-beta pruning alpha-beta pruning. | **04L** |
| UNIT-04 | **Knowledge and Reasoning:** Introduction of knowledge representation and reasoning about objects, relations, events, actions, time, and space; ontologies, frame representation, semantic network, predicate logic, resolution, natural deduction, situation calculus, description logics, reasoning with defaults, reasoning about knowledge. | **06L** |
| UNIT-05 | **Planning:** Methods for planning in advance how to solve a problem, achieving preconditions, backtracking and fixing threats by promotion or demotion: the partial-order planning algorithm, Non-linear planning, Plan Space Planning. | **05L** |
| UNIT-06 | **Uncertain Knowledge and Reasoning:** Probability, connection to logic, independence, certainty factor, Bayes rule, Bayesian networks, probabilistic inference. Dempster-Shafer theory of evidence, Fuzzy logic | **06L** |
| UNIT-07 | **Learning:** Learning from memorization, examples, explanation, and exploration, learning nearest neighbor, naive Bayes, and decision tree classifiers, Q-learning for learning action policies, applications, perceptron. Learning by gradient descent. Multilayer perceptrons and the backpropagation algorithm, Neural Networks. | **05L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Understand the AI and demonstrate awareness of the major challenges facing AI.

CO2: Exhibit strong familiarity with a number of important AI techniques.

CO3: Assess critically the techniques presented and apply them to real world problems.

CO4: Understand the concept of machine learning.

**Books and References**

1. Principles of Artificial Intelligence by N.J. Nilsson, Narosa Publishing House.
2. Artificial Intelligence by Elaine Rich, Kevin Knight and Shivashankar B Nair, McGraw Hill.
3. Artificial Intelligence – A Modern Approach by Stuart Russell and Peter Norvig, Pearson Education.
4. Logic and Prolog Programming by Saroj Kaushik, New Age International.

| Course Name: | **Data warehousing and Data Mining** | |
|---|---|---|
| Course Code: | **CS-421** | |
| Course Type: | **Core** | |
| Contact Hours/Week:  **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about fundamental Data warehousing theory and its architecture.
- To introduce the fundamental concepts demonstrating the competency in the use of CRISP-DM, the Cross-Industry Standard Process for Data Mining.
- To enable the students to understand current data mining techniques and applications, such as text mining, mining genomics data, and other current issues.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** Data warehousing Definition, usage and trends. DBMS vs. Data warehouse, Data marts, Metadata, Multidimensional data mode, Data cubes, Schemas for Multidimensional Database: stars, snowflakes and fact constellations. | **04L** |
| UNIT-02 | **Architecture: Data** warehouse process & architecture, OLTP vs OLAP, ROLAP vs MOLAP, types of OLAP, servers, 3-Tier Data warehouse architecture, distributed and virtual Data warehouses, Data warehouse manager. | **05L** |
| UNIT-03 | **Implementation:** Data warehouse implementation, computation of data cubes, modeling OLAP data, OLAP queries manager, Data warehouse back end tools, complex aggregation at multiple granularities, tuning and testing of Data warehouse. | **06L** |
| UNIT-04 | **Data Mining, Association rules and Classification:** Types of Data, Data Mining Functionalities, Interestingness of Patterns, Classification of Data Mining Systems, Data Mining Task Primitives, Integration of a Data Mining System with a Data warehouse. Issues with Data Preprocessing, Mining Frequent Patterns, Associations and Correlations, Constraint Based Association Mining, Classification and Prediction:  Decision Tree Induction, Bayesian Classification, Rule Based Classification, Classification by Back propagation, Support Vector Machines, Associative Classification, Lazy Learners, Prediction. | **10L** |
| UNIT-05 | **Cluster Analysis:** Cluster Analysis, Types of Data, Categorization of Major Clustering Methods, K-means, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid Based Methods, Model-Based Clustering Methods, Clustering High Dimensional Data, Constraint Based Cluster Analysis, outlier Analysis. | **06L** |
| UNIT-06 | **Other Trends in Data Mining:** Decision tree knowledge discovery through Neural Networks & Genetic Algorithm, Rough Sets and Fuzzy techniques. | **05L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to
CO1:    Describe fundamental data warehousing theory and its architecture.
CO2:    Be able to approach data mining as a process, by demonstrating competency in the use of CRISP-DM, the Cross-Industry Standard Process for Data Mining.
CO3:    Understand and apply a wide range of clustering, estimation, prediction, and classification algorithms.
CO4:    Understand and apply the most current data mining techniques and applications, such as text mining, mining genomics data, and other current issues.

**Books and References**

1. Data Warehousing in the Real World by Sam Anahory and  Dennis Murray, Pearson.
2. Data Mining-Concepts & Techniques by Jiawei Han and  Micheline Kamber, Morgan Kaufmann.
3. Data Mining by Pieter Adriaans and  Dolf Zantinge, Pearson
4. Data Warehousing, Data Mining and OLTP by Alex Berson, McGraw Hill.
5. Developing the Data Warehouses by W. H Inmon and C. Klelly, John Wiley & Sons.
6. Managing the Data Warehouses by W. H. Inmon and  C.L. Gassey, John Wiley & Sons.

| Course Name: | **Information Security** | |
|---|---|---|
| Course Code: | **CS-422** | |
| Course Type: | **Core** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**
- To impart knowledge about the network security to predict and classify attacks on a system/network.
- To introduce the fundamental concepts relevant to Cryptographic Techniques.
- To enable the students to understand security threats to ICT infrastructure using modern tools such as firewalls, UTMs, etc.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** Standards Organizations, Security Components OSI Security Architecture, Aspects of Security, Passive Attacks, Active Attacks, Security Services (X.800), Model for Network Security, Model for Network Access Security, Symmetric Cipher Model, Cryptography Classification, Cryptanalysis, Substitution: Other forms, Poly-alphabetic Substitution Ciphers, One-Time Pad, Transposition (Permutation) Ciphers, Product Ciphers. | **06L** |
| UNIT-02 | **Number Theory and Prime Numbers:** Groups, Rings, and Fields, Modular Arithmetic, Euclid's Algorithm, Finite Fields of the Form GF(p), Polynomial Arithmetic, Finite Fields of the Form $GF(2^n)$. Generation of large prime numbers, Prime factorization, Euler Totient Function $\phi(n)$, Euler's Theorem, Primality Test- Fermat's Little Theorem, Baillie-PSW, Solovay-Strassen, Miller Rabin Algorithm, AKS Algorithm, Cyclotomic primality test, Elliptic Curve Primality Test, Prime Distribution, Chinese Remainder Theorem, Primitive Roots, Discrete Logarithms | **08L** |
| UNIT-03 | **Cryptographic Techniques:** Perfect security, Feistel Cipher Structure, Block Cipher- DES, differential and Linear Cryptanalysis, Avalanche Effect, Double-DES, Triple-DES, Cipher modes of operations: block and stream mode, AES, International Data Encryption Algorithm (IDEA), Blowfish Algorithm; RC4; Pseudo number generation, Blum Blum Shub Generator. | **07L** |
| UNIT-04 | **Public-Key Cryptography and Message Authentication:** The Key Distribution Problem, Public-Key Cryptosystems, The RSA Algorithm, The Key Management riddle, The Diffie-Hellman Key Exchange, Elliptic Curve Cryptography, Message Authentication, requirements and functions, Message Authentication Codes, Hash Functions, Birthday Problem, SHA-X, SHA-512, Authentication, Access control policies, The Message Digest (MD5) Algorithm, HMAC fundamentals, Digital Signature basics, Authentication Protocols, The Digital Signature Standard, Kerberos Authentication scheme, The X.509 Directory Authentication scheme. | **08L** |
| UNIT-05 | **Security Protocols:** Secure User Authentication, Mail security, PGP, database security, File system security, Program security, Memory security, Session security, SSH, Web security, Replay Attacks, Needham Schroeder Protocol, IPSec, SSL, IEEE 802.11, Wired Equivalent Privacy (WEP). | **07L** |

**Course Outcomes**
Upon successful completion of the course, the students will be able to
CO1:    Analyze the basic concepts of network security to predict and classify attacks on a system/network.
CO2:    Illustrate the process of cryptographic algorithms for information hiding.
CO3:    Understand and apply authentication techniques to provide secure communication.
CO4:    Assess the security threats to ICT infrastructure using modern tools such as firewalls, UTMs, etc.

**Books and References**
1. Cryptography and Network Security: Principles and Practice by William Stallings, Pearson Education.
2. Cryptography: Theory and Practice by D Stinson, Chapman & Hall.
3. Network Security by C. Kaufman, R. Perlman and M. Spenser, Prentice Hall of India.
4. Internet Security and Firewalls by S. Bellovin and W. Chesvick, Addison-Wesley, Reading.
5. Introduction to Cryptography with Coding Theory by Wade Trappe and Lawrence C. Washington, Prentice-Hall.
6. Cryptography & Network Security by Behrouz A. Forouzan, McGraw Hill Education.

| Course Name: | **Modelling and Simulation Techniques** | |
|---|---|---|
| Course Code: | **CS-430** | |
| Course Type: | **Professional Elective-I** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- Introduce various system modeling and simulation techniques.
- Highlight their applications in different areas.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Fundamentals:** Definition and reasons for simulation, Continuous (time-oriented) and discrete (event) systems, Modeling/programming simple deterministic systems, Rates and system dynamics. | **07L** |
| UNIT-02 | **Concepts in Simulation**: Stochastic variables, discrete vs continuous probability, Monte Carlo Simulations; Monte Carlo methods, Normally distributed random numbers, Monte Carlo V/S Stochastic Simulations. | **07L** |
| UNIT-03 | **Queuing Models**: Single server queuing system, introduction to arrival and departure time, flowcharts for arrival and departure routine. Event graphs of queuing model. Determining the events and variables, Event graphs for inventory model. | **06L** |
| UNIT-04 | **Random Numbers:** Introduction to Random Numbers, Importance of Random Numbers in Simulation, MidSquare random number generator, Residue method, Arithmetic Congruential generator, Testing Numbers for Randomness, Chi-Square Test | **06L** |
| UNIT-05 | **Discrete Event System Simulation:** Discrete events; representation of time; queues and servers; generation of arrival patterns; resource seizing; departures simulation of a telephone system and computer networks; simulating components of an operating system; delayed calls; modeling policies; priority queues; tasks; gathering statistics; counters and summary statistics; measuring utilization and occupancy; recording distributions and transit times. Introduction to a Simulation Languages Simulation in C++, GPSS/ MATLAB/Network Simulators. | **10L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Gain insight into the operation of a system.

CO2:    Develop operating or resource policies to improve system performance.

CO3:    Test new concepts and/or systems before implementation.

**Books and References**

1.    Discrete Event System Simulation by Jerry Banks and John Carson, PHI.
2.    System Simulation by Geoffrey Gordon, PHI.
3.    System Modeling and Simulation by Frank L. Severance, Wiley.
4.    Simulation Modeling and Analysis by Averill M. Law and W. David Kelton, McGraw Hill.
5.    Handbook of Simulation: Principles by Jerry Banks, Wiley.
6.    Introduction to Probability Models by Sheldon M. Ross, Academic Press.

| Course Name: | **Soft Computing Methods in Engineering** | |
|---|---|---|
| Course Code: | **CS-431** | |
| Course Type: | **Professional Elective-I** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To understand the concepts of feed forward &feedback neural networks.
- To understand the concept of fuzziness involved in various systems.
- To expose the ideas about genetic algorithm.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** Introduction to Neuro-fuzzy and Soft Computing, Fuzzy set theory, Fuzzy Rules, Fuzzy Reasoning, Fuzzy inference System, Neural Networks; Radial basis and recurrent neural networks, Hopfield Networks, Comparison of RBF and MLP Network, Running Algorithms, Neuro-fuzzy Modelling, Applications of Soft Computing to Signal Processing, Image Processing, Forecasting, XOR Problem travelling salesman problem, Image compression suing MLPs character retrieval using Hopfield networks, Introduction to Genetic Algorithm hybrid systems etc. Recent advances in soft computing applications. | **07L** |
| UNIT-02 | Types of learning, supervised and unsupervised learning laws. Learning Laws: Hebb's rule, Delta rule, Widrow - Hoff (The Least-Mean-Square ) learning rule, correlation learning rule, instar and outstar learning rules, Competitive learning, Credit Assignment Problem, Error Correction learning, Memory based learning, Boltzmann learning | **08L** |
| UNIT-03 | The Perceptron and its learning law, Classification of linearly separable patterns, Multi-Layer Perceptron, Supervised Learning, Back-Propagation Learning law. Feed forward networks, Recurrent Networks. | **08L** |
| UNIT-04 | Winner takes-all Networks, Competitive Learning, Kohonen's Self organizing Maps, Introduction to Adaptive Resonance Theory | **08L** |
| UNIT-05 | **GA based Backpropagation Networks**: GA based Weight Determination, K - factor determination in Columns, GA and search space, operators in GA: encoding, selection, crossover and mutation. | **05L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Ability to comprehend machine learning and soft computing techniques in solving real world applications.

CO2:     Ability to design and develop ML techniques.

CO3:     Ability to visualize and analyze behavioural pattern to develop evolutionary algorithm.

**Books and References**

1.    Principles of Soft Computing by S N Sivanandam and S N Deepa, Wiley India.
2.    Soft Computing and Intelligent System Design by Fakhreddine O Karray and  Clarence D Silva. Pearson Edition,
3.    Artificial Intelligence and Soft Computing - Behavioural and Cognitive Modeling of the Human Brain by Amit Konar, CRC Press.
4.    Artificial Intelligence by Patric Henry Winston, Pearson Education.
5.    A first course in Fuzzy Logic by Hung T Nguyen and Elbert A Walker, CRC. Press

| Course Name: | **Data Analytics** | |
|---|---|---|
| Course Code: | **CS-432** | |
| Course Type: | **Professional Elective-I** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To introduce fundamental algorithms and techniques used in Data Analytics.
- To cover statistical foundations followed by various machine learning and data mining algorithms.
- To introduce technological aspects like data management (Hadoop), scalable computation (MapReduce) and visualization.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Data Definitions and Analysis Techniques:** elements, variables, and data categorization, levels of measurement, data management and indexing, introduction to statistical learning and R-programming. | **06L** |
| UNIT-02 | **Descriptive Statistics:** Measures of central tendency, Measures of location of dispersions, Practice and analysis with R. | **06L** |
| UNIT-03 | **Basic Analysis Techniques:** Basic analysis techniques, Statistical hypothesis generation and testing, Chi-Square test, t-Test, Analysis of variance, Correlation analysis, Maximum likelihood test, Practice and analysis with R. | **07L** |
| UNIT-04 | **Data Analysis Techniques:** Regression analysis, Classification techniques, Clustering, Association rules analysis. | **07L** |
| UNIT-05 | **Case Studies and Projects:** Understanding business scenarios, Feature engineering and visualization, Scalable and parallel computing with Hadoop and Map-Reduce, Sensitivity Analysis. | **10L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Find a meaningful pattern in data and graphically interpret data

CO2: Implement the analytic algorithms

CO3: Handle large scale analytics projects from various domains

CO4: Develop intelligent decision support systems

**Books and References**

1. Probability and Statistics for Engineers & Scientists By Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying Ye, Prentice Hall.
2. The Elements of Statistical Learning, Data Mining, Inference, and Prediction by Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer.
3. An Introduction to Statistical Learning: with Applications in R by G James, D. Witten, T Hastie, and R. Tibshirani, Springer.
4. Mining Massive Data Sets by Jure Leskovec, Anand Rajaraman and Jeff Ullman, Cambridge University Press.

| Course Name: | **Cyber Forensics** | |
|---|---|---|
| Course Code: | **CS-450** | |
| Course Type: | **Professional Elective-II** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about forensics and cyber laws.
- To introduce collection of evidence.
- To impart knowledge about host, network and multimedia forensics.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction to Forensics:** Introduction to Incident, First Responder Procedure, Incident Response Methodology, Investigation Steps, Incident Handling, Investigation Reports, Setup of Cyber Forensics Lab, Cyber forensics Readiness, Introduction to Cyber Laws, Anti-forensics techniques. | **07L** |
| UNIT-02 | **Evidence Collection and Handling:** Evidence Collection from Computer, Smartphones, IoT, Cloud. Disk Imaging- Tools, Evidence Preservation, Write Blockers, Chain of Custody, Challenges in evidence handling, Understanding Filesystems - Windows, Linux, Android, Concept of Slack space. | **07L** |
| UNIT-03 | **Host Forensics:** Memory Forensics, Malware Analysis, Reverse Engineering Tools, Encryption, Password Cracking, Rainbow tables, Recovery of deleted files, File carving | **06L** |
| UNIT-04 | **Network Forensics:** Introduction to network protocols, Network packet analysis, Collecting Network Based Evidence, Network Intrusion detection, Investigating Routers, Email Tracing, Internet Fraud. Dark Web, TOR network, Application of Big Data techniques for Log Analysis. | **06L** |
| UNIT-05 | **Multimedia Forensics and Case Studies:** Image Forensics, Video Forensics, Audio Forensics, Steganography, Social Media Forensics, Identity theft, Corporate espionage, Online Defamation, Online harassment, mobile forensics, memory forensics. | **10L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Understand concepts of forensic analysis.

CO2:    Perform image, video and audio forensics.

CO3:    Identify possibility of tampering with the forensic evidences.

**Books and References**

1.    Digital Evidence and Computer Crime by Eoghan Casey, Academic Press.
2.    Real Digital Forensics: Computer Security and Incident Response by Keith J. Jones, Richard Bejtlich and Curtis Wayne Rose, Addison-Wesley.
3.    File system forensic analysis by Brian Carrier, Addison-Wesley Professional.
4.    Digital Forensics by André Årnes, Wiley.
5.    A Practical Guide to Computer Forensics Investigations by Darren R. Hayes, Pearson IT Certification.

| Course Name: | **Wireless Sensor Networks** | |
|---|---|---|
| Course Code: | **CS-451** | |
| Course Type: | **Professional Elective-II** | |
| Contact Hours/Week:  **3L** | | Course Credits: **03** |

**Course Objectives**

- Understand various components of a sensor node and discuss various applications of WSNs.
- Compare and evaluate routing protocols in wireless sensor networks.
- Understand various topology control algorithms in WSNs.
- Compare and contrast various data processing techniques used in wireless sensor networks.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Sensor networks overview:** Introduction, applications, design issues, requirements. Sensor node architecture, Architecture and factors influencing the sensor network design. | **07L** |
| UNIT-02 | **Network architecture:** Optimization goals, evaluation metrics, network design principles. Sensor network operating systems and brief introduction to sensor network programming. | **07L** |
| UNIT-03 | **Network protocols:** <br> MAC protocols and energy efficiency. | **07L** |
| UNIT-04 | **Routing protocols:** Data centric, hierarchical, location-based, energy efficient routing etc. Sensor deployment, scheduling and coverage issues, self configuration and topology control. | **07L** |
| UNIT-05 | Querying, data collection and processing, collaborative information processing and group connectivity, Target tracking, localization and identity management. Power management. Security and privacy. | **08L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Have an understanding of the principles and characteristics of wireless sensor networks.
CO2:     Apply knowledge of wireless sensor networks to various application areas.
CO3:     Analyze WSN protocols in terms of their energy efficiency and design new energy efficient protocols.

**Books and References**

1.    Wireless Sensor Networks-An Information Processing Approach by Feng Zhao and Leonidas Guibas, Morgan Kauffman.
2.    Wireless Sensor Networks: From Theory to Applications  by  Ibrahiem M. M. El Emary and S. Ramakrishnan, CRC Press.
3.    Wireless sensor networks by Edgar H. Callaway, AUERBACH Publications.
4.    Wireless Sensor Networks: Principles and Practice  by Fei Hu and Xiaojun Cao, CRC Press.
5.    Protocols and Architectures for Wireless Sensor Networks by Andreas Willig and Holger Karl, Wiley Publications.

| Course Name: | **Next Generation Networks** | |
|---|---|---|
| Course Code: | **CS-452** | |
| Course Type: | **Professional Elective-II** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about the Next Generation Networks, Resource sharing and design issues for the network layers.
- To introduce the fundamental concepts related to the process, file system and storage management.
- To enable the students to understand the Cloud, Fog and Edge Computing; MANETS, VANETs, WSNs; Cognitive Radio Networks; Internet of things; Software Defined Networking (SDN).

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | Introduction to Next Generation Networks, Resource sharing, Design issues for the network layers Technologies influencing networking, wide area wireless networks, Networking devices | **07L** |
| UNIT-02 | Principles of internetworking, Tunneling, Fragmentation, Naming and addressing concepts, Hierarchical naming, Domain name system, Name resolution process | **07L** |
| UNIT-03 | IP address classes and concept of subnetting, Classless Inter-domain routing (CIDR) and DHCP concepts, The internet protocols: IP, ICMP, ARP, RARP | **07L** |
| UNIT-04 | The internet protocols: TCP and UDP, IP Everywhere, Wireline and Wireless networking, VoIP, VoLTE, Mobility in networks, Mobile IP; IoT: Introduction to IoT, Time for Convergence, IoT Strategic Research and Innovation Directions, IoT Applications, Future Internet Technologies, IoT Related Standardization. | **07L** |
| UNIT-05 | Introduction to Cloud, Fog and Edge Computing; MANETS, VANETs, WSNs; Cognitive Radio Networks; IoT; Software Defined Networking (SDN). | **08L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Implement and realize various privileged operations critical to the functioning of an operating system and network.
CO2:     Understand and apply various operations related to the process, file system and storage management.
CO3:     Understand various addressing and sub-netting techniques, routing protocols, etc.
CO4:     Use and configure TCP/IP protocol for network setup.
CO5:     Appreciate legal and ethical issues related to system and network security.

**Books and References**

1.   Next Generation Network Services by Robert Wood, Pearson.
2.   Next Generation Network Services, Technologies and Strategies by Neill Wilkinson, Wiley.
3.   Data Communications and Networking by B.A. Forouzan, McGraw Hill.
4.   Computer Networks by A.S. Tanenbaum, Prentice Hall of India.
5.   Introduction to Wireless and Mobile Systems by Dharma P. Agrawal and Q-An Zeng, Thomson Learning Inc.

| Course Name: | **Information Retrieval** | |
|---|---|---|
| Course Code: | **CS-440** | |
| Course Type: | **Professional Elective-III** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To become familiar with difference between Information retrieval and data base management systems.
- Students will be able to learn different indexing techniques to apply information retrieval systems
- Students will be able to understand various searching techniques to retrieve information from information retrieval systems.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction to Information Retrieval:** Information retrieval process, Information Retrieval vs. Data Retrieval, Indexing, Information retrieval model, Boolean retrieval model. Dictionary and Postings: Tokenization, Stop words, Stemming, Inverted index, Skip pointers, Phrase queries. Query Languages for IR: Keywords, Boolean Queries, Context Queries, Natural Language Queries, Structural Queries. | **07L** |
| UNIT-02 | **Tolerant Retrieval:** Wild card queries, General Trees, Permuterm index, Bigram index, Ngrams, redux; spelling correction: Edit distance, Jaccard coefficient, Ngrams, re-redux; Phonetics - Soundex. Term Weighting and Vector Space Model: Wild card queries, Permuterm index, Bigram index, spelling correction, Edit distance, Jaccard coefficient, Soundex. Evaluation Precision: Recall, F-measure, E-measure, Normalized recall, Evaluation problems. | **07L** |
| UNIT-03 | **Latent Semantic Indexing:** Eigen vectors, Singular value decomposition, Lowrank approximation, Problems with Lexical Semantics. Query Expansion: Query Expansion and its types, Query drift, Relevance feedback, Rocchio algorithm, Probabilistic relevance feedback, Automatic Local Analysis, Automatic Global Analysis, Document Clustering. | **07L** |
| UNIT-04 | **Probabilistic Information Retrieval:** Probabilistic relevance feedback, Probability ranking principle, Binary Independence Model, Bayesian network for text retrieval. XML Indexing and Search: Data vs. Text-centric XML, Text-Centric XML retrieval, Structural terms, Knuth-Morris-Pratt, Boyer-Moore family, Suffix automaton, Phrases and Proximity, Search Advertising, duplicate detection. | **07L** |
| UNIT-05 | **Multimedia Information Retrieval:** Similarity Queries, Feature-based Indexing and Searching, Spatial Access Methods, Searching in Multidimensional Spaces. Meta-Ranking: Integrated vs. Isolated Methods, Interleaving, Voting. Web Search: History of Web, Indexing, Spidering/Crawling, Link Analysis (HITS, Page Rank), User Interfaces and Visualization. Search evaluation, retrieval feedback, search log mining, and applications in web information management | **08L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to
CO1:    Identify various information retrieval systems.
CO2:    Use knowledge of data structures and indexing methods in information retrieval Systems.
CO3:    Choose clustering and searching techniques for different information retrieval systems.
CO4:    Explain and evaluate different types of search engines.

**Books and References**

1.    Introduction to Information Retrieval by C. Manning, P. Raghavan, and H. Schutze, Cambridge University Press.
2.    Modern Information Retrieval: The Concepts and Technology behind Search by Ricardo Baeza Yates and Berthier Ribeiro Neto, ACM Press Books.
3.    Search Engines: Information Retrieval by W.Bruce Croft, Donald Metzler and Trevor Strohman, Addison Wesley.
4.    An Introduction to Search Engines and Web Navigation by Mark Levene, Wiley.
5.    Information Retrieval Implementing and Evaluating Search Engines by Stefan Büttcher, Charles L. A. Clarke and Gordon V. Cormack, MIT Press.
6.    Modern Information Retrieval by Baeza-Yates and Ribeiro-Neto, Addison Wesley.

| Course Name: | **Humanoid Robotics** | |
|---|---|---|
| Course Code: | **CS-441** | |
| Course Type: | **Professional Elective-III** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To understand the concepts of robotics
- To understand the concept of kinematics, optimization and control in robotics
- To design robotics system for real applications

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** **O**verview, robot mechanisms, characteristics, Types of Robots, Scope and limitations of robots, Advantages, Disadvantages, Applications. | **07L** |
| UNIT-02 | **General Robot Architecture:** Understanding the mapping between Robots configurations space and Cartesian/ working space, Forward and inverse kinematics, Homogeneous coordinate-transformation principle, Mathematical model of a robot, introductory robot dynamics, Trajectory planning vs learning. Real time master slave architecture for robot control, Robot sensors with emphasis on vision sensor. | **08L** |
| UNIT-03 | **Robot Vision and Control:** Multiple images and fundamentals of stereo vision, Human Robot Interactions through Gesture and speech recognition using HMM Human Robot Interactions through Gesture and speech recognition using deep learning techniques such as CNN and RNN. | **08L** |
| UNIT-04 | **Cognitive robotics:** Creating intelligent behaviors using purely reactive approach through subsumption Architecture, Potential field based architecture, Potential field based architecture, deliberative approach and hybrid deliberative and reactive approach. | **08L** |
| UNIT-05 | **Intelligent robot grasping:** Grasp pose estimation, Grasp planning vs learning, applications of deep learning techniques in robot grasping. <br> **On Robot mobility:** Object detection and semantic segmentation for self-driving cars/drone, outdoor SLAM (Simultaneous Localization and Mapping). | **05L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

  CO1: Understand and describe the basic properties of robotics.

  CO2: Derive and apply the kinematic equations for a basic robot system.

  CO3: Understand and discuss technical challenges with robotics

**Books and References**

1. Robot Modeling and Control by Mark W. Spong, Seth Hutchinson and M. Vidyasagar, Wiley.
2. Modern Robotics: Mechanics, Planning, and Control by Kevin M. Lynch and Frank C. Park, Cambridge University Press.
3. Introduction to Robotics by John J. Craig, Addison-Wesley Publishing.
4. Mobile Robotics: Mathematics, Models, and Methods, Alonzo Kelly, Cambridge University Press.
5. Robotics, Vision and Control by Peter Corke, Springer.

| Course Name: | **Natural Language Processing** | |
|---|---|---|
| Course Code: | **CS-442** | |
| Course Type: | **Professional Elective-III** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To understand the application of computational methods in linguists.
- To apply statistical and probabilistic methods for parameter estimation and inference.
- To know how the computational methods give insight into observed human language phenomena.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** Knowledge in speech and language processing, Ambiguity, Models and Algorithms, Brief History Regular Expressions and Automata: Regular Expressions, Finite-State Automata, Regular Languages and FSA Morphology and Transducers: Inflectional and derivational morphology, finite state morphological parsing, Combining FST Lexicon and rules. Lexicon free FST: Porter Stemmer N-grams: Counting Words in Corpora, SIMPLE (UNSMOOTHED) N-GRAMS, Smoothing. | **08L** |
| UNIT-02 | **Word Classes and Part-of-Speech Tagging:** English word classes, Targets for English, Part of speech Tagging, Rule Based part of speech Tagging, Transformation Based Tagging. Context Free Grammars for English: Constituency, Context Free rules and Trees, Sentence level construction, The Noun Phrase, Coordination, Agreement, The verb phrase and sub-categorization. Spoken Language Syntax, Grammar Equivalence and Normal form, Finite state context free grammars, Grammar and human processing. Problems with probabilistic context free grammars, probabilistic lexicalized GFG. | **07L** |
| UNIT-03 | **Parsing with context free grammars**: Parsing as Search, Basic Top down Parser, Problems with basic top-down-parsers, the early Algorithm, Finite state parsing method Features and Unifications: Feature structures, Unification of Features Structures, Features Structures in the grammar, Implementing Unification. Lexicalized and probabilistic parsing: Probabilistic context free grammars. | **07L** |
| UNIT-04 | S**emantics Representing Meaning:** Computational Desiderata for representation, Meaning structure of language, First order predicate calculus, linguistically relevant concept, Related Re-presentational approaches, Alternative approaches to meaning. Semantic Analysis: Syntax driven semantic analysis, Attachment of Fragment of English, Integrating semantic analysis with early parser. Robust Semantic Analysis. Lexical Semantics: Relation among lexemes and their senses, Internal Structure of words. | **07L** |
| UNIT-05 | **Pragmatics Discourse:** Reference resolution, Text Coherence, Discourse Structure, Psycholinguistics Studies of reference and coherence. Natural Language generation: Introduction to language generation, Architecture for generation, Surface realization, Discourse planning, Macro planning, Lexical selection, evaluating generation systems, generating speech. | **07L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Compare and contrast natural language processing approaches.

CO2:    Comprehend and analyze various elements of speech processing.

CO3:    Design and develop machine learning techniques in the area of NLP.

**Books and References**

1.    Speech and Language Processing by Daniel Jurafsky and James H. Martin, Prentice Hall.
2.    NLP: A Paninian Perspective by Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, Prentice Hall.
3.    Language as a Cognitive Process by T. Winograd, Addison-Wesley.

| Course Name: | **Reconfigurable Computing** | |
|---|---|---|
| Course Code: | **CS-460** | |
| Course Type: | **Professional Elective-IV** | |
| Contact Hours/Week:  **3L** | | Course Credits: **03** |

**Course Objectives**

- To understand various computing architectures.
- To provide students the concept of handling issues of reconfigure computing.
- To provide students implementation approaches of FPGA design in view of reconfiguration.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | Domain of RC: General Purpose Computing, Domain-Specific Processors, Application-Specific Processors, Reconfigurable Computing, Fields of Application. Architecture of Field Programmable Gate Arrays, Reconfigurable Processing Fabric (RPF) Architectures: Fine grained, Coarse-Grained, Integration of RPF into Traditional Computing Systems | **08L** |
| UNIT-02 | Early systems of Reconfigurable computing: PAM, VCC, Splash, PRISM, Teramac, Cray, SRC, non-FPGA research, other issues; Reconfiguration Management: Reconfiguration, Configuration architectures, managing reconfiguration process, reducing reconfiguration time, configuration security | **08L** |
| UNIT-03 | Implementation: Integration, FPGA Design Flow, System On A Programmable Chip: Introduction to SoPC, Adaptive Multiprocessing on Chip. | **06L** |
| UNIT-04 | Reconfiguration Project Design Approaches: J-Bit, Modular, Early Access, Vivado. | **06L** |
| UNIT-05 | RC Applications: Implementing applications with FPGAs, various applications and use of reconfiguration: Video Streaming, Distributed arithmetic, Adaptive Controller, Adaptive cryptographic systems, Software Defined Radio, High-Performance Computing, Automatic target recognition systems | **08L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to.

CO1:   Understand the concept of reconfigurable computing and its integration on computing platform.

CO2:   Design, implement and analyze reconfigurable systems in the recent application domains using HDL.

CO3:   Use advanced EDA tools to simulate and synthesize HDL codes for reconfigurable architectures.

**Books and References**

1.   Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation  by Scott Hauck, André DeHon.
2.   CMOS VLSI Design by N.H.E.Weste, D. Harris, Pearson
3.   FPGA- based System Design by W. Wolf, Pearson.
4.   Reconfigurable computing: the theory and practice of FPGA-based computation by S.Hauck, A.DeHon, Elsevier.

| Course Name: | **Cloud Computing** | |
|---|---|---|
| Course Code: | **CS-461** | |
| Course Type: | **Professional Elective-IV** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To provide comprehensive knowledge of fundamental concepts of cloud computing.
- To demonstrate an understanding of Service models, deployment models, Virtualization.
- To describe the programming and software environments of Cloud.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction to Cloud Computing:** Roots of Cloud Computing, Layers and Types of Clouds, Features of a Cloud, Cloud Infrastructure Management, Infrastructure as a Service Providers, Platform as a Service Providers, and Challenges and Opportunities**.** | **07L** |
| UNIT-02 | **Virtualization and Resource Provisioning in Clouds:** Introduction and Inspiration, Virtual Machines (VM), VM Provisioning and Manageability, VM Migration Services, VM Provisioning in the Cloud Context, and Future Research Directions. | **07L** |
| UNIT-03 | **Cloud Computing Architecture:** Cloud Benefits and Challenges, Market-Oriented Cloud Architecture, SLA-oriented Resource Allocation, Global Cloud Exchange; Emerging Cloud Platforms, Federation of Clouds | **07L** |
| UNIT-04 | **Programming Enterprise Clouds using Aneka:** Introduction, Aneka Architecture, Aneka Deployment, Parallel Programming Models, Thread Programming using Aneka, Task Programming using Aneka, and MapReduce Programming using Aneka, Parallel Algorithms, Parallel Data mining, Parallel Mandelbrot, and Image Processing. | **07L** |
| UNIT-05 | **Advanced Topics and Cloud Applications:** Integration of Private and Public Clouds, Cloud Best Practices, GrepTheWeb on Amazon Cloud, ECG Data Analysis on Cloud using Aneka, Hosting Massively Multiplayer Games on Cloud, and Content Delivery Networks Using Clouds, and Hosting Twitter and Facebook on Cloud. | **08L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:     Identify roots of cloud computing and fundamental concepts of cloud computing.

CO2:     Describe Programming Enterprise Clouds using Aneka.

CO3:     Apply principles of programming and software environments of Cloud.

**Books and References**

1.   Cloud Computing: Principles and Paradigms by R. Buyya, J. Broberg and A. Goscinski, Wiley Press,
2.   Cloud Computing: A Practical Approach by A.T. Velte, T. J. Velte, Robert Elsenpeter, McGraw-Hill.
3.   Cloud Computing Bible by B. Sosinsky, Wiley Publishing.
4.   Architecting the Cloud: Design Decisions for Cloud Computing Service Models by M. J. Kavis, Wiley Publishing.

| Course Name: | **Advanced Computer Architecture** | |
|---|---|---|
| Course Code: | **CS-462** | |
| Course Type: | **Professional Elective-IV** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about the quantitative principles of computer architecture.
- To introduce the fundamental concepts relevant to analyze the ILP and associated hazards.
- To enable the students to understand superscalar, vector and VLIW styles of computing for various applications.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** review of basic computer architecture, quantitative techniques in computer design, measuring and reporting performance. CISC and RISC processors. Pipelining: Basic concepts, instruction and arithmetic pipeline, data hazards, control hazards, and structural hazards, techniques for handling hazards. | **07L** |
| UNIT-02 | **Exception handling:** Pipeline optimization techniques. Compiler techniques for improving performance. Hierarchical memory technology: Inclusion, Coherence and locality properties. | **07L** |
| UNIT-03 | **Cache memory organizations:** Techniques for reducing cache misses; Virtual memory organization, mapping and management techniques, memory replacement policies. | **07L** |
| UNIT-04 | **Instruction-level parallelism:** basic concepts, techniques for increasing ILP, superscalar, superpipelined and VLIW processor architectures. Array and vector processors. | **07L** |
| UNIT-05 | **Multiprocessor architecture:** taxonomy of parallel architectures. Centralized shared-memory architecture: synchronization, memory consistency, interconnection networks. Distributed shared-memory architecture. Cluster computers. Non von Neumann architectures: data flow computers, reduction computer architectures, systolic architectures. | **08L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1: Understand quantitative principles of computer architecture.
CO2: Understand and analyze the ILP and associated hazards.
CO3: Design memory hierarchies for better performance.
CO4: Identify superscalar, vector and VLIW styles of computing for various applications.

**Books and References**

1. Computer Organization and Design: A Hardware/Software Interface by John L. Hennessy and David A. Patterson, Morgan Kaufmann.
2. Computer Architecture: A Quantitative Approach by John L. Hennessy and David A. Patterson, Morgan Kaufmann.
3. Advanced Computer Architecture: Parallelism, Scalability, Programmability by Kai Hwang, McGraw-Hill.

| Course Name: | **Data Structures** | |
|---|---|---|
| Course Code: | **CS-306** | |
| Course Type: | **Open Elective-I / II** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about linear and non-linear data structures as the foundational base for computer solutions to problems.
- To introduce the fundamental concepts relevant to binary trees, binary tree traversals, binary search trees and perform related analysis to solve problems.
- To enable the students to understand various types of sorting algorithms.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introduction:** Data types, data structures, abstract data types, the running time of a program, the running time and storage cost of algorithms, complexity, asymptotic complexity, big O notation, obtaining the complexity of an algorithm. | **07L** |
| UNIT-02 | **Development of Algorithms:** Notations and Analysis, Storage structures for arrays - sparse matrices - structures and arrays of structures, Stacks and Queues: Representations, implementations and applications. Linked Lists: Singly linked lists, Linked stacks and queues, operations on Polynomials, Doubly Linked Lists, Circularly Linked Lists, Operations on linked lists. | **10L** |
| UNIT-03 | **Trees:** Basic terminology, General Trees, Binary Trees, Tree Traversing: in-order, pre-order and post-order traversal, building a binary search tree, Operations on Binary Trees. | **07L** |
| UNIT-04 | **Graphs:** Basic definitions, representations of directed and undirected graphs, the single-source shortest path problem, the all-pair shortest path problem, traversals of directed and undirected graphs, directed acyclic graphs, strong components, minimum cost spanning tress, articulation points and biconnected components, graph matching. | **06L** |
| UNIT-05 | **Sorting and Searching Techniques:** Bubble sorting, Insertion sort, Selection sort, Shell sort, Merge sort, Heap and Heap sort, Quick sort, Sequential searching, Binary Searching, Index searching, Hash table methods. | **06L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Interpret and compute asymptotic notations of an algorithm to analyze the time complexity.

CO2:    Use of linear and non-linear data structures as the foundational base for computer solutions to problems.

CO3:    Demonstrate the ability to implement various types of static and dynamic lists.

CO4:    Implement binary trees, binary tree traversals, and binary search trees and perform related analysis to solve problems.

CO5:    Implement various types of sorting algorithms.

**Books and References**

1.    An Introduction to Data Structures with applications by J.P. Tremblay and P.G. Sorenson, Tata McGraw Hill.
2.    Data structures, Algorithms ad Applications in C++ by Sartaj Sahni, WCB/McGraw Hill.
3.    Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman and John E. Hopcroft, Addison Wesley.
4.    Data Structures using C by Y. Langsam, M. J. Augenstein and A. M. Tenenbaum, Pearson Education.
5.    Data Structures – A Pseudocode Approach with C by Richard F. Gilberg and Behrouz A. Forouzan, Thomson Brooks /COLE.

| Course Name: | **Operating System** | |
|---|---|---|
| Course Code: | **CS-370** | |
| Course Type: | **Open Elective-I** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about the concepts of operating system and its management.
- To introduce the fundamental concepts scheduling of processes for a given problem instance.
- To enable the students to understand memory management techniques and implement replacement algorithms and understand and implement file systems.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Evolution of Operating Systems:** Evolution of operating systems, Types of operating systems. The process concept, system programmer's view of processes, operating system's views of processes, operating system services for process management. | **05L** |
| UNIT-02 | **CPU Scheduling:** Scheduling concepts, scheduling algorithms, algorithm evaluation, multiple processor scheduling, real time scheduling. | **06L** |
| UNIT-03 | **Concurrent Programming and Deadlocks:** Critical regions, Conditional critical regions, Monitors, Interprocess communication, Messages, Pipes, Semaphores, Modularization, Synchronization, Concurrent languages. Deadlocks: Characterization, Prevention, Avoidance, Detection and Recovery, Combined approach to Deadlock Handling, precedence graphs. | **06L** |
| UNIT-04 | **Memory Management:** Memory Management, Contiguous allocation, static-swapping, overlays, dynamic partitioned memory allocation, demand paging, page replacement, segmentation. Non-contiguous allocation, paging, Hardware support, Virtual Memory. | **07L** |
| UNIT-05 | **File Systems:** A Simple file system, General model of a file system, Symbolic file system, Access control verification, Logical file system, Physical file system, Allocation strategy module, Device strategy module, I/O initiators, Device handlers, Disk scheduling. | **06L** |
| UNIT-06 | **Networks, Security and Design Principles:** Network operating system, distributed operating system, external security, operational security. | **06L** |

**Course Outcomes**

Upon successful completion of the course, the students will be able to

CO1:    Understand and analyze the concepts of operating system and its management.

CO2:    Illustrate the scheduling of processes for a given problem instance.

CO3:    Identify the dead lock situation and provide appropriate solution.

CO4:    Analyze memory management techniques and implement replacement algorithms.

CO5:    Understand and implement file systems.

**Books and References**

1.    Operating System Concepts by J.L. Peterson and A. Silberchatz, Addison Wesley.
2.    An Introduction to Operating System by Harvey M. Dietel, Addison Wesley.
3.    Operating Systems - A Design Oriented Approach by C. Crowley, Irwin Publishing.
4.    Operating systems by W. Stallings, Prentice Hall.
5.    Modern Operating system by A.S. Tanenbaum, Prentice Hall of India.

| Course Name: | **Computer Networks** | |
|---|---|---|
| Course Code: | **CS-380** | |
| Course Type: | **Open Elective-II** | |
| Contact Hours/Week: **3L** | | Course Credits: **03** |

**Course Objectives**

- To impart knowledge about the network models and architectures.
- To introduce the fundamental concepts relevant to performance of various routing protocols and design of new routing protocol.
- To enable the students to understand computers, software, networking technologies and information assurance to an organization's management, operations, and requirements.

| Unit Number | Course Content | Lectures |
|---|---|---|
| UNIT-01 | **Introductory Concepts:** Goals and Applications of Networks, LAN, WAN, MAN, Wireless network, Network software: Protocol hierarchies, design issues of layers, Interfaces and services. Reference Model: The OSI reference model, TCP/IP reference model, Example networks: Novell Netware, The ARPANET, The Internet, X-25 Networks, network standards. | **05L** |
| UNIT-02 | **Physical Layer:** Fourier Analysis, Maximum data rate of a channel, Transmission media, Wireless transmission, Narrowband ISDN, Broadband ISDN and ATM, Virtual circuits, Circuit switching, Communication satellite. | **04L** |
| UNIT-03 | **Data Link Layer:** Data link layer design issues, services provided to network layers, Framing, Error control, Flow control, Error detection and correction, Elementary data link protocols, An unrestricted Simplex protocol, A Simplex Stop-and-Wait protocol, Simplex Protocol for a noisy channel, Sliding Window protocols, A one-bit Sliding protocol, A protocol using go-back-N, A protocol using selective repeat, Protocol specification and verification, Example data link protocol-HDLC, PPP and SLIP. | **06L** |
| UNIT-04 | **Medium Access Sublayer:** Channel Allocations, Static and dynamic allocation in LAN and MAN, Multiple Access protocols, ALOHA, Carrier Sense multiple access protocols, WDMA protocols, Wireless protocols, Collision free protocols, Limited contention protocols, IEEE standard 802.3 and Ethernet, IEEE standard 802.4, Token bus IEEE standard 802.5, Token Ring, Distributed Queue Dual bus, Logical link control, bridges, High speed LAN, Satellite networks. | **06L** |
| UNIT-05 | **Network Layer:** Network Layer design issue, Routing algorithms, Congestion Control Algorithms, Internetworking. **Transport Layer:** Transport services, Design issues, elements of transport protocols, simple transport protocols, Connection management, TCP, UDP. | **10L** |
| UNIT-06 | **Session, Presentation and Application Layer:** Session Layer, Design issues, remote procedure call. Presentation Layer, Design issues, Data compression techniques, cryptography. Application Layer - File Transfer, Access and Management, Electronic mail, Virtual Terminals, Other applications, Example Networks - Internet and Public Networks. | **05L** |

**Course Outcomes**
Upon successful completion of the course, the students will be able to
CO1:    Understand network models and architectures.
CO2:    Identify the pros and cons of choosing a suitable MAC layer protocol.
CO3:    Analyze the performance of various routing protocols and design of new routing protocol.
CO4:    Solve basic network design problems using knowledge of common local and wide area network architectures.

**Books and References**
1.    Computer Networks by A.S. Tanenbaum, Prentice Hall of India.
2.    Computer Networking: A Top-Down Approach Featuring the Internet by J. Kurose and K.W. Ross, Addison-Wesley.
3.    Data and Computer Communication by W. Stallings, Prentice Hall of India.